Rotation Notation/Convention

Commented [JJ1]: We now force NumFlaps >= 1 and NumPylons >= 1 in KiteAeroDyn

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = [\Lambda] \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} \hat{x}^T \\ \hat{y}^T \\ \hat{z}^T \end{bmatrix} \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \text{ (from global to local)}$$

or equivalently:

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = [\Lambda]^T \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} == \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

where *X*/*Y*/*Z* are global coordinates, *x*/*y*/*z* are local coordinates, Λ is the DCM from global to local, and $\hat{x} / \hat{y} / \hat{z}$ are the unit vectors of the local coordinate system expressed in the global coordinate system.

$$\begin{Bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} = F^{EulerExtract}\left(\left[\Lambda\left(\theta_x, \theta_y, \theta_z\right) \right] \right)$$

where function $F^{EulerExtract}(\;)$ returns the 3 Euler angles of the *x*-*y*-*z* (1-2-3) rotation sequence used to form Λ (that is, first a rotation θ_x about the global *X* axis, followed by rotation θ_y about the *Y'* axis, followed by rotation θ_z about the *Z''* axis) defined as follows:

$$\Lambda\left(\theta_x, \theta_y, \theta_z\right) = \begin{bmatrix} COS(\theta_z) & SIN(\theta_z) & 0 \\ -SIN(\theta_z) & COS(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} COS(\theta_y) & 0 & -SIN(\theta_y) \\ 0 & 1 & 0 \\ SIN(\theta_y) & 0 & COS(\theta_y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & COS(\theta_x) & SIN(\theta_x) \\ 0 & -SIN(\theta_x) & COS(\theta_x) \end{bmatrix}$$

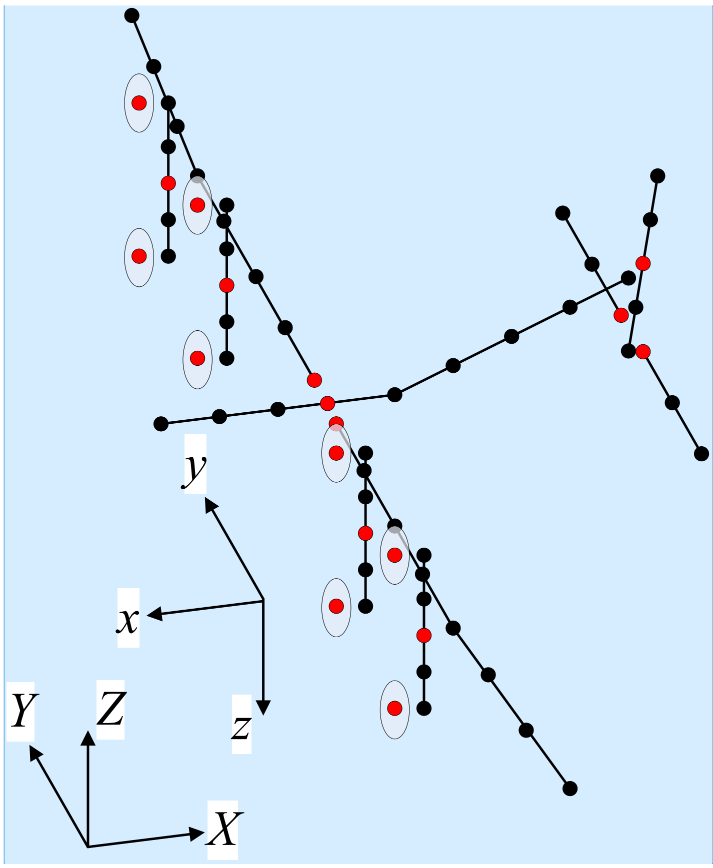
$$= \begin{bmatrix} COS(\theta_y)COS(\theta_z) & COS(\theta_x)SIN(\theta_z) + SIN(\theta_x)SIN(\theta_y)COS(\theta_z) & SIN(\theta_x)SIN(\theta_z) - COS(\theta_x)SIN(\theta_y)COS(\theta_z) \\ -COS(\theta_y)SIN(\theta_z) & COS(\theta_x)COS(\theta_z) - SIN(\theta_x)SIN(\theta_y)SIN(\theta_z) & SIN(\theta_x)COS(\theta_z) + COS(\theta_x)SIN(\theta_y)SIN(\theta_z) \\ SIN(\theta_y) & -SIN(\theta_x)COS(\theta_y) & COS(\theta_x)COS(\theta_y) \end{bmatrix}$$

Note the following simplifications:

$$\Lambda\left(0, \theta_y, \theta_z\right) = \begin{bmatrix} COS(\theta_y)COS(\theta_z) & SIN(\theta_z) & -SIN(\theta_y)COS(\theta_z) \\ -COS(\theta_y)SIN(\theta_z) & COS(\theta_z) & SIN(\theta_y)SIN(\theta_z) \\ SIN(\theta_y) & 0 & COS(\theta_y) \end{bmatrix}$$

$$\Lambda\left(\theta_x, 0, \theta_z\right) = \begin{bmatrix} COS(\theta_z) & COS(\theta_x)SIN(\theta_z) & SIN(\theta_x)SIN(\theta_z) \\ -SIN(\theta_z) & COS(\theta_x)COS(\theta_z) & SIN(\theta_x)COS(\theta_z) \\ 0 & -SIN(\theta_x) & COS(\theta_x) \end{bmatrix}$$

$$\Lambda\left(\theta_x, \theta_y, 0\right) = \begin{bmatrix} COS(\theta_y) & SIN(\theta_x)SIN(\theta_y) & -COS(\theta_x)SIN(\theta_y) \\ 0 & COS(\theta_x) & SIN(\theta_x) \\ SIN(\theta_y) & -SIN(\theta_x)COS(\theta_y) & COS(\theta_x)COS(\theta_y) \end{bmatrix}$$



Commented [JJ2]: This orientation of the energy kite is shown as KiteRoll=0, KitePitch=180, KiteYaw=0.

KiteAeroDyn Driver

KiteAeroDyn Driver Input File

The user specifies the general configuration of the energy kite in the KiteAeroDyn driver input file. The general configuration assumes:

- one fuselage
- two wings (starboard, port) attached to the fuselage
- one vertical stabilizer attached to the fuselage
- two horizontal stabilizers (starboard, port) attached to the vertical stabilizer
- a user-specified number of pylons attached to each wing
- two rotors (top, bottom) per pylon
- nominal circular motion of the energy kite in the port (left) direction

The general configuration of the control surfaces assumes:

- a user-specified number of flaps per wing
- two rudders on the vertical stabilizer
- two elevators per horizontal stabilizer

- variable rotor speed
- variable rotor-collective blade-pitch angles

The number of pylons/rotors, as well as the undeflected reference point (origin-red nodes in figure above) of each of these energy kite components is specified in a body-fixed (x,y,z) coordinate system in the KiteAeroDyn driver, with x pointed forward (in the primary direction of flight), y pointed starboard (right) (when looking in the primary direction of flight), and z pointed down (following the right-hand rule). (The fuselage reference point is taken to be (0,0,0).) When KiteAeroDyn is coupled within KiteFAST, the reference points are specified within the KiteMBDyn preprocessor. These reference points enable proper spatial mesh-to-mesh mapping between the aerodynamics computed by KiteAeroDyn and the structural dynamics computed by MBDyn. The reference points are added for convenience in defining the geometry of the energy kite; the coordinates of all reference points may be set to (0,0,0) if desired.

The user also specifies the ambient horizontal wind speed, including direction and vertical shear, the global rigid-body time-history motion (translational and rotational positions and velocities) and control settings (rotor speeds, rotor collective blade-pitch angles, and flap, rudder, and elevator control settings) of the energy kite. The wind and global motion of the energy kite are specified in a global inertial-frame (X,Y,Z) coordinate system, with X pointed in the nominal 0˚ wind direction, Z pointed up (opposite gravity), and Y pointed to the left when looking downwind along 0˚ wind (following the right-hand rule).

Flow chart

Initialization:

Read-in the input file data from the Kite AeroDyn driver input file, and:

- Convert angular units of input parameters from degrees to radians.
- Check inputs and set parameters

Call KiteAeroDyn_Init()

Open Output File

Set the reference positions and orientations of the fuselage reference point mesh:

$\vec{p}^{FusOR} = \vec{0}$;

$\Lambda^{FusOR} = I$

Set mesh-mappings between the fuselage reference point mesh and the KiteAeroDyn input meshes.

$n = 0$

$t = 0$

Initial calculate output:

Set inputs to KiteAeroDyn at $t = 0$ (see below for solution at t)

Call KiteAeroDyn_CalcOutput()

Write Output to File

Time increment:

Set the displacement, orientation, and velocities of the fuselage reference point mesh based on the specified motion at $t = t + \Delta t$ (used in the mesh-mappings below):

- $\vec{u}^{FusO} = \begin{Bmatrix} INTERP1D\left(KitePxi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KitePyi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KitePzi\left[:\right],Time\left[:\right]@t\right) \end{Bmatrix}$
- $\Lambda^{FusO} = \Lambda\left(INTERP1D\left(KiteRoll\left[:\right],Time\left[:\right]@t\right),INTERP1D\left(KitePitch\left[:\right],Time\left[:\right]@t\right),INTERP1D\left(KiteYaw\left[:\right],Time\left[:\right]@t\right)\right)$
- $\vec{v}^{FusO} = \begin{Bmatrix} INTERP1D\left(KiteTVxi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KiteTVyi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KiteTVzi\left[:\right],Time\left[:\right]@t\right) \end{Bmatrix}$
- $\vec{\omega}^{FusO} = \frac{\pi}{180}\begin{Bmatrix} INTERP1D\left(KiteRVxi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KiteRVyi\left[:\right],Time\left[:\right]@t\right) \\ INTERP1D\left(KiteRVzi\left[:\right],Time\left[:\right]@t\right) \end{Bmatrix}$

Set inputs to KiteAeroDyn at t:

- $\vec{u}_j^{Fus} = M_u^{P2L}\left(\vec{u}^{FusO},\Lambda^{FusO}\right)$
- $\Lambda_j^{Fus} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{Fus} = M_v^{P2L}\left(\vec{u}_j^{Fus},\vec{u}^{FusO},\vec{v}^{FusO},\vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{Fus} = Wind\left(\left\{{}^{In}\vec{p}_j^{FusR}+\vec{u}_j^{Fus}\right\}\bullet\hat{Z}\right)$

  Note: The $Wind\left(Z\right)$ function used in this equation and several equations below is as follows:

$$Wind\left(Z\right) = \begin{Bmatrix} HWindSpd\left(\frac{Z}{Re\,fHt}\right)^{PLexp} COS\left(HWindDir\right) \\ -HWindSpd\left(\frac{Z}{Re\,fHt}\right)^{PLexp} SIN\left(HWindDir\right) \\ 0 \end{Bmatrix}$$

- $\vec{u}_j^{SWn} = M_u^{P2L}\left(\vec{u}^{FusO},\Lambda^{FusO}\right)$
- $\Lambda_j^{SWn} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{SWn} = M_v^{P2L}\left(\vec{u}_j^{SWn},\vec{u}^{FusO},\vec{v}^{FusO},\vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{SWn} = Wind\left(\left\{{}^{In}\vec{p}_j^{SWnR}+\vec{u}_j^{SWn}\right\}\bullet\hat{Z}\right)$
- $\vec{u}_j^{PWn} = M_u^{P2L}\left(\vec{u}^{FusO},\Lambda^{FusO}\right)$
- $\Lambda_j^{PWn} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{PWn} = M_v^{P2L}\left(\vec{u}_j^{PWn},\vec{u}^{FusO},\vec{v}^{FusO},\vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{PWn} = Wind\left(\left\{{}^{In}\vec{p}_j^{PWnR}+\vec{u}_j^{PWn}\right\}\bullet\hat{Z}\right)$

- $\vec{u}_j^{VS} = M_u^{P2L}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda_j^{VS} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{VS} = M_v^{P2L}\left(\vec{u}_j^{VS}, \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{VS} = Wind\left(\left\{{}^{In}\vec{p}_j^{VSR} + \vec{u}_j^{VS}\right\} \bullet \hat{Z}\right)$
- $\vec{u}_j^{SHS} = M_u^{P2L}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda_j^{SHS} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{SHS} = M_v^{P2L}\left(\vec{u}_j^{SHS}, \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{SHS} = Wind\left(\left\{{}^{In}\vec{p}_j^{SHSR} + \vec{u}_j^{SHS}\right\} \bullet \hat{Z}\right)$
- $\vec{u}_j^{PHS} = M_u^{P2L}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda_j^{PHS} = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{PHS} = M_v^{P2L}\left(\vec{u}_j^{PHS}, \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{PHS} = Wind\left(\left\{{}^{In}\vec{p}_j^{PHSR} + \vec{u}_j^{PHS}\right\} \bullet \hat{Z}\right)$
- $\vec{u}_j^{SPy}\left[n_{Pylons}\right] = M_u^{P2L}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda_j^{SPy}\left[n_{Pylons}\right] = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{SPy}\left[n_{Pylons}\right] = M_v^{P2L}\left(\vec{u}_j^{SPy}\left[n_{Pylons}\right], \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{SPy}\left[n_{Pylons}\right] = Wind\left(\left\{{}^{In}\vec{p}_j^{SPyR}\left[n_{Pylons}\right] + \vec{u}_j^{SPy}\left[n_{Pylons}\right]\right\} \bullet \hat{Z}\right)$
- $\vec{u}_j^{PPy}\left[n_{Pylons}\right] = M_u^{P2L}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda_j^{PPy}\left[n_{Pylons}\right] = M_\Lambda^{P2L}\left(\Lambda^{FusO}\right)$
- $\vec{v}_j^{PPy}\left[n_{Pylons}\right] = M_v^{P2L}\left(\vec{u}_j^{PPy}\left[n_{Pylons}\right], \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}_j^{PPy}\left[n_{Pylons}\right] = Wind\left(\left\{{}^{In}\vec{p}_j^{PPyR}\left[n_{Pylons}\right] + \vec{u}_j^{PPy}\left[n_{Pylons}\right]\right\} \bullet \hat{Z}\right)$
- $\vec{u}^{SPyRtr}\left[n_{Pylons}, n_2\right] = M_u^{P2P}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda^{SPyRtr}\left[n_{Pylons}, n_2\right] = M_\Lambda^{P2P}\left(\Lambda^{FusO}\right)$
- $\vec{v}^{SPyRtr}\left[n_{Pylons}, n_2\right] = M_v^{P2P}\left(\vec{u}^{SPyRtr}\left[n_{Pylons}, n_2\right], \vec{u}^{FusO}, \vec{v}^{FusO}, \vec{\omega}^{FusO}\right)$
- $\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right] = Wind\left(\left\{\vec{p}^{SPyRtrR}\left[n_{Pylons}, n_2\right] + \vec{u}^{SPyRtr}\left[n_{Pylons}, n_2\right]\right\} \bullet \hat{Z}\right)$
- $\Omega^{SPyRtr}\left[n_{Pylons}, n_2\right] = INTERP1D\left(SP\left\langle n_{Pylons}\right\rangle\left\langle T, B\right\rangle RtSpd\left[:\right], Time\left[:\right]@t\right)$
- $\theta^{SPyRtr}\left[n_{Pylons}, n_2\right] = INTERP1D\left(SP\left\langle n_{Pylons}\right\rangle\left\langle T, B\right\rangle Pitch\left[:\right], Time\left[:\right]@t\right)$
- $\vec{u}^{PPyRtr}\left[n_{Pylons}, n_2\right] = M_u^{P2P}\left(\vec{u}^{FusO}, \Lambda^{FusO}\right)$
- $\Lambda^{PPyRtr}\left[n_{Pylons}, n_2\right] = M_\Lambda^{P2P}\left(\Lambda^{FusO}\right)$

- $\vec{v}^{PPyRtr}\left[n_{Pylons},n_2\right]=M_v^{P2P}\left(\vec{u}^{PPyRtr}\left[n_{Pylons},n_2\right],\vec{u}^{FusO},\vec{v}^{FusO},\vec{\omega}^{FusO}\right)$
- $\vec{V}^{PPyRtr}\left[n_{Pylons},n_2\right]=Wind\left(\left\{\vec{p}^{PPyRtrR}\left[n_{Pylons},n_2\right]+\vec{u}^{PPyRtr}\left[n_{Pylons},n_2\right]\right\}\bullet\hat{Z}\right)$
- $\Omega^{PPyRtr}\left[n_{Pylons},n_2\right]=INTERP1D\left(PP\left\langle n_{Pylons}\right\rangle\left\langle T,B\right\rangle RtSpd\left[:\right],Time\left[:\right]@t\right)$
- $\theta^{PPyRtr}\left[n_{Pylons},n_2\right]=INTERP1D\left(PP\left\langle n_{Pylons}\right\rangle\left\langle T,B\right\rangle Pitch\left[:\right],Time\left[:\right]@t\right)$
- $Ctrl^{SFlp}\left[n_{Flaps}\right]=INTERP1D\left(SFlp\left\langle n_{Flaps}\right\rangle Ctrl\left[:\right],Time\left[:\right]@t\right)$
- $Ctrl^{PFlp}\left[n_{Flaps}\right]=INTERP1D\left(PFlp\left\langle n_{Flaps}\right\rangle Ctrl\left[:\right],Time\left[:\right]@t\right)$
- $Ctrl^{Rudr}\left[n_2\right]=INTERP1D\left(Rudr\left\langle 1,2\right\rangle Ctrl\left[:\right],Time\left[:\right]@t\right)$
- $Ctrl^{SElv}\left[n_2\right]=INTERP1D\left(SElv\left\langle 1,2\right\rangle Ctrl\left[:\right],Time\left[:\right]@t\right)$
- $Ctrl^{PElv}\left[n_2\right]=INTERP1D\left(PElv\left\langle 1,2\right\rangle Ctrl\left[:\right],Time\left[:\right]@t\right)$

Call KiteAeroDyn_UpdateStates()

$n=n+1$
$t=t+\Delta t$

Call KiteAeroDyn_CalcOutput()

Write Output to File

End:
Call KiteAeroDyn_End

Close Output File

KiteAeroDyn

Primary KiteAeroDyn Input File

The undeflected aerodynamic geometry of the components of the energy kite are defined in the primary KiteAeroDyn input file. (Likewise, the undeflected structural geometry of the components of the energy kite are defined in the input file for the KiteMBDyn preprocessor.) For the fuselage, wings, vertical and horizontal stabilizers, and pylons, the aerodynamic nodes define the aerodynamic center of an airfoil (reference point for the airfoil lift and drag forces, which is assumed to be the ¼ chord location within the KiteVSM submodel). The aerodynamic reference line and aerodynamic loads are assumed to be piecewise linear between each node (to establish step changes in loads between nodes—e.g. between nodes with and without control surfaces—would require placing two nodes very close each other).

Each component is specified as follows:

- *Fuselage*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with x monotonically increasing (from possibly negative to positive values). The airfoil at each node is assumed to be in the y-z plane, and—along with the nodal locations—the positive aerodynamic twist is specified about positive x, and the chordlength and airfoil table ID are specified. A zero-degree twist means positive y points toward the trailing edge and negative z points toward the suction side of the airfoil.
- *Starboard (right) wing*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with y monotonically increasing. The airfoil at each node is assumed to be rotated from the x-z plane based on the dihedral angle about negative x resulting in an inclined x-z' plane (with y' normal), and—along with the nodal locations—the positive aerodynamic twist is specified about positive y', and the chordlength, airfoil table ID, and flap ID are specified. A zero-degree twist means negative x points toward the trailing edge and negative z' points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating airfoil data, the airfoil and flap IDs at each midpoint is taken to be the airfoil and flap IDs of the corresponding node with lower y.
- *Port (left) wing*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with y monotonically decreasing (negative values). The airfoil at each node is assumed to be rotated from the x-z plane based on the dihedral angle about positive x resulting in an inclined x-z' plane (with y' normal), and—along with the nodal locations—the positive aerodynamic twist is specified about positive y', and the chordlength, airfoil table ID, and flap ID are specified. A zero-degree twist means negative x points toward the trailing edge and negative z' points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating airfoil data, the airfoil and flap IDs at each midpoint is taken to be the airfoil and flap IDs of the corresponding node with higher (less negative) y.
- Vertical stabilizer: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with z monotonically increasing (from possible negative to positive values). The airfoil at each node is assumed to be in the x-y plane, and—along with the nodal locations—the positive aerodynamic twist is specified about positive z, and the chordlength, airfoil table ID, and rudder ID are specified. A zero-degree twist means negative x points toward the trailing edge and positive y points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating airfoil data, the airfoil and rudder IDs at each midpoint is taken to be the airfoil and rudder IDs of the corresponding node with lower z.
- *Starboard (right) horizontal stabilizer*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with y monotonically increasing. The airfoil at each node is assumed to be in the x-z plane, and—along with the nodal locations—the positive aerodynamic twist is specified about positive y, and the chordlength, airfoil table ID, and elevator ID are specified. A zero-degree twist means negative x points toward the trailing edge and negative z points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating

Commented [JJ3]: Not true, could also be monotonically decreasing, but A and B are sent to VSM in order (A is used to control the airfoil); also for output
Collocated nodes (i.e. the same x for the fuselage) result in no VSM element (for step changes) (even if y and z are different, there is no element created if the x is the same)

airfoil data, the airfoil and elevator IDs at each midpoint is taken to be the airfoil and elevator IDs of the corresponding node with lower y.

- *Port (left) horizontal stabilizer*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with y monotonically decreasing (negative values). The airfoil at each node is assumed to be in the x-z plane, and—along with the nodal locations—the positive aerodynamic twist is specified about positive y, and the chordlength, airfoil table ID, and elevator ID are specified. A zero-degree twist means negative x points toward the trailing edge and negative z points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating airfoil data, the airfoil and elevator IDs at each midpoint is taken to be the airfoil and elevator IDs of the corresponding node with higher (less negative) y.
- *Pylons*: The locations of the aerodynamic nodes (black nodes in figure above) along the aerodynamic reference line (¼ chord) are specified in the body-fixed (x,y,z) coordinate system relative to its origin, with z monotonically increasing (from possibly negative to positive values). The airfoil at each node is assumed to be in the x-y plane, and—along with the nodal locations—the positive aerodynamic twist is specified about positive z, and the chordlength and airfoil table ID are specified. A zero-degree twist means negative x points toward the trailing edge and positive y points toward the suction side of the airfoil. Calculations for the lifting line vortex method take place at the midpoints between these nodes; instead of interpolating airfoil data, the airfoil ID at each midpoint is taken to be the airfoil ID of the corresponding node with lower z.
- *Rotors*: Each rotor is in the y-z plane and the origin is coalescent with the hub center. The rotor radius and actuator disk input file are specified.

Inputs	Outputs	States	Parameters
• $\vec{u}^{FusO}$ – Translational displacement (relative) of the fuselage reference point mesh (m) • $\vec{u}_j^{Fus}$ – Translational displacement (relative) of the j th node of the fuselage mesh (m) • Λ_j^{Fus} – Displaced rotation (absolute orientation) of the j th node of the fuselage mesh (-) • $\vec{v}_j^{Fus}$ – Translational velocity (absolute) of the j th node of the fuselage mesh (m/s) • $\vec{V}_j^{Fus}$ – Undisturbed wind speed computed at the j th node of the fuselage i.e. at ${}^{In}\vec{p}_j^{FusR}+\vec{u}_j^{Fus}$ (m/s) • $\vec{u}_j^{SWn}$ – Translational displacement (relative) of the j th node of the starboard wing mesh (m) • Λ_j^{SWn} – Displaced rotation (absolute orientation) of the j th node of the starboard wing mesh (-) • $\vec{v}_j^{SWn}$ – Translational velocity (absolute) of the j th node of the starboard wing	• ${}^{Out}\vec{u}_j^{Fus}$ – Translational displacement (relative) of the j th node of the fuselage mesh (m) • ${}^{Out}\Lambda_j^{Fus}$ – Displaced rotation (absolute orientation) of the j th node of the fuselage mesh (-) • $\vec{F}_j^{Fus}$ – Aerodynamic applied concentrated forces at the j th node of the fuselage mesh (N) • $\vec{M}_j^{Fus}$ – Aerodynamic applied concentrated moments at the j th node of the fuselage mesh (N-m) • ${}^{Out}\vec{u}_j^{SWn}$ – Translational displacement (relative) of the j th node of the starboard wing mesh		• N_{Flaps} – Number of flaps per wing (-) • N_{Pylons} – Number of pylons per wing (-) • ${}^{In}\vec{p}_j^{FusR}$ – Reference position of the j th node of the fuselage input mesh (m) • ${}^{In}\Lambda_j^{FusR}$ – Reference orientation of the j th node of the fuselage input mesh (-) • ${}^{In}\vec{p}_j^{SWnR}$ – Reference position of the j th node of the starboard wing input mesh (m) • ${}^{In}\Lambda_j^{SWnR}$ – Reference orientation of the j th node of the starboard wing input mesh (-) • ${}^{In}\vec{p}_j^{PWnR}$ – Reference position of the j th node of the port wing input mesh (m)

Commented [JJ4]: VSM has discrete states.

Commented [jmj5]: Not all parameters are listed here. E.g. DTAero, UseCm, AirDens, etc. are obvious…

Commented [JJ6]: Note that the translational displacements and orientations are fields added to the output load (force and moment) meshes for convenience in the moment-related mesh mappings because the input and output meshes differ.

mesh (m/s) • $\vec{V}_j^{SWn}$ – Undisturbed wind speed computed at the j th node of the starboard wing i.e. at ${}^{In}\vec{p}_j^{SWnR}+\vec{u}_j^{SWn}$ (m/s) • $\vec{u}_j^{PWn}$ – Translational displacement (relative) of the j th node of the port wing mesh (m) • Λ_j^{PWn} – Displaced rotation (absolute orientation) of the j th node of the port wing mesh (-) • $\vec{v}_j^{PWn}$ – Translational velocity (absolute) of the j th node of the port wing mesh (m/s) • $\vec{V}_j^{PWn}$ – Undisturbed wind speed computed at the j th node of the port wing i.e. at ${}^{In}\vec{p}_j^{PWnR}+\vec{u}_j^{PWn}$ (m/s) • $\vec{u}_j^{VS}$ – Translational displacement (relative) of the j th node of the vertical stabilizer mesh (m) • Λ_j^{VS} – Displaced rotation (absolute orientation) of the j th node of the vertical stabilizer mesh (-) • $\vec{v}_j^{VS}$ – Translational velocity (absolute) of the j th node of the vertical stabilizer mesh (m/s) • $\vec{V}_j^{VS}$ – Undisturbed wind speed computed at the j th node of the vertical stabilizer i.e. at ${}^{In}\vec{p}_j^{VSR}+\vec{u}_j^{VS}$ (m/s) • $\vec{u}_j^{SHS}$ – Translational displacement (relative) of the j th node of the starboard horizontal stabilizer mesh (m) • Λ_j^{SHS} – Displaced rotation (absolute orientation) of the j th node of the starboard horizontal stabilizer mesh (-) • $\vec{v}_j^{SHS}$ – Translational velocity (absolute) of the j th node of the starboard horizontal stabilizer mesh (m/s) • $\vec{V}_j^{SHS}$ – Undisturbed wind speed computed at the j th node of the starboard horizontal stabilizer i.e. at	(m) • ${}^{Out}\Lambda_j^{SWn}$ – Displaced rotation (absolute orientation) of the j th node of the starboard wing mesh (-) • $\vec{F}_j^{SWn}$ – Aerodynamic applied concentrated forces at the j th node of the starboard wing mesh (N) • $\vec{M}_j^{SWn}$ – Aerodynamic applied concentrated moments at the j th node of the starboard wing mesh (N-m) • ${}^{Out}\vec{u}_j^{PWn}$ – Translational displacement (relative) of the j th node of the port wing mesh (m) • ${}^{Out}\Lambda_j^{PWn}$ – Displaced rotation (absolute orientation) of the j th node of the port wing mesh (-) • $\vec{F}_j^{PWn}$ – Aerodynamic applied concentrated forces at the j th node of the port wing mesh (N) • $\vec{M}_j^{PWn}$ – Aerodynamic applied concentrated moments at the j th node of the port wing mesh (N-m) • ${}^{Out}\vec{u}_j^{VS}$ – Translational displacement (relative) of the j th node of the vertical stabilizer mesh (m) • ${}^{Out}\Lambda_j^{VS}$ – Displaced rotation (absolute orientation) of the j th		• ${}^{In}\Lambda_j^{PWnR}$ – Reference orientation of the j th node of the port wing input mesh (-) • ${}^{In}\vec{p}_j^{VSR}$ – Reference position of the j th node of the vertical stabilizer input mesh (m) • ${}^{In}\Lambda_j^{VSR}$ – Reference orientation of the j th node of the vertical stabilizer input mesh (-) • ${}^{In}\vec{p}_j^{SHSR}$ – Reference position of the j th node of the starboard horizontal stabilizer input mesh (m) • ${}^{In}\Lambda_j^{SHSR}$ – Reference orientation of the j th node of the starboard horizontal stabilizer input mesh (-) • ${}^{In}\vec{p}_j^{PHSR}$ – Reference position of the j th node of the port horizontal stabilizer input mesh (m) • ${}^{In}\Lambda_j^{PHSR}$ – Reference orientation of the j th node of the port horizontal stabilizer input mesh (-) • ${}^{In}\vec{p}_j^{SPyR}\left[n_{Pylons}\right]$ – Reference position of the j th node of the pylons on the starboard wing input mesh (m) • ${}^{In}\Lambda_j^{SPyR}\left[n_{Pylons}\right]$ – Reference orientation of the j th node of the pylons on the starboard wing input mesh (-)

${}^{In}\vec{p}_j^{SHSR}+\vec{u}_j^{SHS}$ (m/s) • $\vec{u}_j^{PHS}$ – Translational displacement (relative) of the j th node of the port horizontal stabilizer mesh (m) • Λ_j^{PHS} – Displaced rotation (absolute orientation) of the j th node of the port horizontal stabilizer mesh (-) • $\vec{v}_j^{PHS}$ – Translational velocity (absolute) of the j th node of the port horizontal stabilizer mesh (m/s) • $\vec{V}_j^{PHS}$ – Undisturbed wind speed computed at the j th node of the port horizontal stabilizer i.e. at ${}^{In}\vec{p}_j^{PHSR}+\vec{u}_j^{PHS}$ (m/s) • $\vec{u}_j^{SPy}\left[n_{Pylons}\right]$ – Translational displacement (relative) of the j th node of the pylons on the starboard wing mesh (m) • $\Lambda_j^{SPy}\left[n_{Pylons}\right]$ – Displaced rotation (absolute orientation) of the j th node of the pylons on the starboard wing mesh (-) • $\vec{v}_j^{SPy}\left[n_{Pylons}\right]$ – Translational velocity (absolute) of the j th node of the pylons on the starboard wing mesh (m/s) • $\vec{V}_j^{SPy}\left[n_{Pylons}\right]$ – Undisturbed wind speed computed at the j th node of the pylons on the starboard wing i.e. at ${}^{In}\vec{p}_j^{SPyR}\left[n_{Pylons}\right]+\vec{u}_j^{SPy}\left[n_{Pylons}\right]$ (m/s) • $\vec{u}_j^{PPy}\left[n_{Pylons}\right]$ – Translational displacement (relative) of the j th node of the pylons on the port wing mesh (m) • $\Lambda_j^{PPy}\left[n_{Pylons}\right]$ – Displaced rotation (absolute orientation) of the j th node of the pylons on the port wing mesh (-) • $\vec{v}_j^{PPy}\left[n_{Pylons}\right]$ – Translational velocity (absolute) of the j th node of the pylons on the port wing mesh (m/s) • $\vec{V}_j^{PPy}\left[n_{Pylons}\right]$ – Undisturbed wind speed computed at the j th node of the pylons on the port wing i.e. at	node of the vertical stabilizer mesh (-) • $\vec{F}_j^{VS}$ – Aerodynamic applied concentrated forces at the j th node of the vertical stabilizer mesh (N) • $\vec{M}_j^{VS}$ – Aerodynamic applied concentrated moments at the j th node of the vertical stabilizer mesh (N-m) • ${}^{Out}\vec{u}_j^{SHS}$ – Translational displacement (relative) of the j th node of the starboard horizontal stabilizer mesh (m) • ${}^{Out}\Lambda_j^{SHS}$ – Displaced rotation (absolute orientation) of the j th node of the starboard horizontal stabilizer mesh (-) • $\vec{F}_j^{SHS}$ – Aerodynamic applied concentrated forces at the j th node of the starboard horizontal stabilizer mesh (N) • $\vec{M}_j^{SHS}$ – Aerodynamic applied concentrated moments at the j th node of the starboard horizontal stabilizer mesh (N-m) • ${}^{Out}\vec{u}_j^{PHS}$ – Translational displacement (relative) of the j th node of the port horizontal stabilizer mesh (m) • ${}^{Out}\Lambda_j^{PHS}$ – Displaced rotation (absolute orientation) of the j th node of the port horizontal stabilizer		• ${}^{In}\vec{p}_j^{PPyR}\left[n_{Pylons}\right]$ – Reference position of the j th node of the pylons on the port wing input mesh (m) • ${}^{In}\Lambda_j^{PPyR}\left[n_{Pylons}\right]$ – Reference orientation of the j th node of the pylons on the port wing input mesh (-) • ${}^{Out}\vec{p}_j^{FusR}$ – Reference position of the j th node of the fuselage output mesh (m) • ${}^{Out}\Lambda_j^{FusR}$ – Reference orientation of the j th node of the fuselage output mesh (-) • ${}^{Out}\vec{p}_j^{SWnR}$ – Reference position of the j th node of the starboard wing output mesh (m) • ${}^{Out}\Lambda_j^{SWnR}$ – Reference orientation of the j th node of the starboard wing output mesh (-) • ${}^{Out}\vec{p}_j^{PWnR}$ – Reference position of the j th node of the port wing output mesh (m) • ${}^{Out}\Lambda_j^{PWnR}$ – Reference orientation of the j th node of the port wing output mesh (-) • ${}^{Out}\vec{p}_j^{VSR}$ – Reference position of the j th node of the vertical stabilizer output mesh (m) • ${}^{Out}\Lambda_j^{VSR}$ – Reference orientation of the j th node of the vertical stabilizer output mesh (-)

${}^{In}\vec{p}_j^{PPyR}\left[n_{Pylons}\right]+\vec{u}_j^{PPy}\left[n_{Pylons}\right]$ (m/s) • $\vec{u}^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Translational displacement (relative) of the top and bottom rotors on the pylons on the starboard wing mesh (m) • $\Lambda^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Displaced rotation (absolute orientation) of the top and bottom rotors on the pylons on the starboard wing mesh (-) • $\vec{v}^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Translational velocity (absolute) of the top and bottom rotors on the pylons on the starboard wing mesh (m/s) • $\vec{V}^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Undisturbed wind speed computed at the top and bottom rotors on the pylons on the starboard wing i.e. at $\vec{p}^{SPyRtrR}\left[n_{Pylons},n_2\right]+\vec{u}^{SPyRtr}\left[n_{Pylons},n_2\right.$ (m/s) • $\Omega^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Rotational speeds of the top and bottom rotors on the pylons on the starboard wing (rad/s) • $\theta^{SPyRtr}\left[n_{Pylons},n_2\right]$ – Rotor-collective blade-pitch angles of the top and bottom rotors on the pylons on the starboard wing (rad) • $\vec{u}^{PPyRtr}\left[n_{Pylons},n_2\right]$ – Translational displacement (relative) of the top and bottom rotors on the pylons on the port wing mesh (m) • $\Lambda^{PPyRtr}\left[n_{Pylons},n_2\right]$ – Displaced rotation (absolute orientation) of the top and bottom rotors on the pylons on the port wing mesh (-) • $\vec{v}^{PPyRtr}\left[n_{Pylons},n_2\right]$ – Translational velocity (absolute) of the top and bottom rotors on the pylons on the port wing mesh (m/s) • $\vec{V}^{PPyRtr}\left[n_{Pylons},n_2\right]$ – Undisturbed wind speed computed at the top and bottom rotors on the pylons on the port wing i.e. at $\vec{p}^{PPyRtrR}\left[n_{Pylons},n_2\right]+\vec{u}^{PPyRtr}\left[n_{Pylons},n_2\right.$ (m/s)	mesh (-) • $\vec{F}_j^{PHS}$ – Aerodynamic applied concentrated forces at the j^{th} node of the port horizontal stabilizer mesh (N) • $\vec{M}_j^{PHS}$ – Aerodynamic applied concentrated moments at the j^{th} node of the port horizontal stabilizer mesh (N-m) • ${}^{Out}\vec{u}_j^{SPy}\left[n_{Pylons}\right]$ – Translational displacement (relative) of the j^{th} node of the pylons on the starboard wing mesh (m) • ${}^{Out}\Lambda_j^{SPy}\left[n_{Pylons}\right]$ – Displaced rotation (absolute orientation) of the j^{th} node of the pylons on the starboard wing mesh (-) • $\vec{F}_j^{SPy}\left[n_{Pylons}\right]$ – Aerodynamic applied concentrated forces at the j^{th} node of the pylons on the starboard wing mesh (N) • $\vec{M}_j^{SPy}\left[n_{Pylons}\right]$ – Aerodynamic applied concentrated moments at the j^{th} node of pylons on the starboard wing mesh (N-m) • ${}^{Out}\vec{u}_j^{PPy}\left[n_{Pylons}\right]$ – Translational displacement (relative) of the j^{th} node of the pylons on the port wing mesh (m) • ${}^{Out}\Lambda_j^{PPy}\left[n_{Pylons}\right]$ –		• ${}^{Out}\vec{p}_j^{SHSR}$ – Reference position of the j^{th} node of the starboard horizontal stabilizer output mesh (m) • ${}^{Out}\Lambda_j^{SHSR}$ – Reference orientation of the j^{th} node of the starboard horizontal stabilizer output mesh (-) • ${}^{Out}\vec{p}_j^{PHSR}$ – Reference position of the j^{th} node of the port horizontal stabilizer output mesh (m) • ${}^{Out}\Lambda_j^{PHSR}$ – Reference orientation of the j^{th} node of the port horizontal stabilizer output mesh (-) • ${}^{Out}\vec{p}_j^{SPyR}\left[n_{Pylons}\right]$ – Reference position of the j^{th} node of the pylons on the starboard wing output mesh (m) • ${}^{Out}\Lambda_j^{SPyR}\left[n_{Pylons}\right]$ – Reference orientation of the j^{th} node of the pylons on the starboard wing output mesh (-) • ${}^{Out}\vec{p}_j^{PPyR}\left[n_{Pylons}\right]$ – Reference position of the j^{th} node of the pylons on the port wing output mesh (m) • ${}^{Out}\Lambda_j^{PPyR}\left[n_{Pylons}\right]$ – Reference orientation of the j^{th} node of the pylons on the port wing output mesh (-) • $\vec{p}^{SPyRtrR}\left[n_{Pylons},n_2\right]$ – Reference positions (origins) of the top and bottom rotors on the

• $\Omega^{PPyRtr}\left[n_{Pylons}, n_2\right]$ – Rotational speeds of the top and bottom rotors on the pylons on the port wing (rad/s) • $\theta^{PPyRtr}\left[n_{Pylons}, n_2\right]$ – Rotor-collective blade-pitch angles of the top and bottom rotors on the pylons on the port wing (rad) • $Ctrl^{SFlp}\left[n_{Flaps}\right]$ – Flap control settings on the starboard wing (user) • $Ctrl^{PFlp}\left[n_{Flaps}\right]$ – Flap control settings on the port wing (user) • $Ctrl^{Rudr}\left[n_2\right]$ – Rudder control settings on the vertical stabilizer (user) • $Ctrl^{SElv}\left[n_2\right]$ – Elevator control settings on the starboard horizontal stabilizer (user) • $Ctrl^{PElv}\left[n_2\right]$ – Elevator control settings on the port horizontal stabilizer (user)	Displaced rotation (absolute orientation) of the j th node of the pylons on the port wing mesh (-) • $\vec{F}_j^{PPy}\left[n_{Pylons}\right]$ – Aerodynamic applied concentrated forces at the j th node of the pylons on the port wing mesh (N) • $\vec{M}_j^{PPy}\left[n_{Pylons}\right]$ – Aerodynamic applied concentrated moments at the j th node of pylons on the port wing mesh (N-m) • $\vec{F}^{SPyRtr}\left[n_{Pylons}, n_2\right]$ – Aerodynamic applied concentrated forces at the top and bottom rotors on the pylons on the starboard wing mesh (N) • $\vec{M}^{SPyRtr}\left[n_{Pylons}, n_2\right]$ – Aerodynamic applied concentrated moments at the top and bottom rotors on the pylons on the starboard wing mesh (N-m) • $\vec{F}^{PPyRtr}\left[n_{Pylons}, n_2\right]$ – Aerodynamic applied concentrated forces at the top and bottom rotors on the pylons on the port wing mesh (N) • $\vec{M}^{PPyRtr}\left[n_{Pylons}, n_2\right]$ – Aerodynamic applied concentrated moments at the top and bottom rotors on the pylons on the port wing mesh (N-m)		pylons on the starboard wing input/output mesh (m) • $\Lambda^{SPyRtrR}\left[n_{Pylons}, n_2\right]$ – Reference orientations of the top and bottom rotors on the pylons on the starboard wing input/output mesh (-) • $\vec{p}^{PPyRtrR}\left[n_{Pylons}, n_2\right]$ – Reference positions (origins) of the top and bottom rotors on the pylons on the port wing input/output mesh (m) • $\Lambda^{PPyRtrR}\left[n_{Pylons}, n_2\right]$ – Reference orientations of the top and bottom rotors on the pylons on the port wing input/output mesh (-) • $LiftMod$ – Lifting-line calculation model (switch) • $RotorMod$ – Rotor calculation model (switch)

Initialization:

Initialization Inputs	Initialization Outputs
• KAD_InFile – Name of the primary KiteAeroDyn input file (string) • N_{Flaps} – Number of flaps per wing (-) • N_{Pylons} – Number of pylons per wing (-) • $\vec{p}^{SWnOR}$ – Reference position (origin) of the starboard wing (m) • $\vec{p}^{PWnOR}$ – Reference position (origin) of the port wing (m) • $\vec{p}^{VSOR}$ – Reference position (origin) of the vertical stabilizer (m) • $\vec{p}^{SHSOR}$ – Reference position (origin) of the starboard horizontal stabilizer (m) • $\vec{p}^{PHSOR}$ – Reference position (origin) of the port horizontal stabilizer (m) • $\vec{p}^{SPyOR}\left[n_{Pylons}\right]$ – Reference positions (origins) of pylons on the starboard wing (m) • $\vec{p}^{PPyOR}\left[n_{Pylons}\right]$ – Reference positions (origins) of pylons on the port wing (m) • $\vec{p}^{SPyRtrR}\left[n_{Pylons},n_2\right]$ – Reference positions (origins) of the top and bottom rotors on the pylons on the starboard wing (m) • $\vec{p}^{PPyRtrR}\left[n_{Pylons},n_2\right]$ – Reference positions (origins) of the top and bottom rotors on the pylons on the port wing (m)	• ρ – Air density (kg/m^3)

Commented [JJ7]: Greg also added nIfWPts, the number of points where inflow wind is needed.

Note that:

Set parameters from initialization inputs (N_{Flaps} and N_{Pylons}). Note that:

- The flap indices: $n_{Flaps} = \left\{1,2,\ldots,N_{Flaps}\right\}$
- The pylon indices: $n_{Pylons} = \left\{1,2,\ldots,N_{Pylons}\right\}$
- And: $n_2 = \left\{1,2\right\}$

Read-in the input file data from KAD_InFile, and:

- Convert angular units of input parameters from degrees to radians.
- Verify that $FusX_j$, $SWnY_j$, VSZ_j, $SHSY_j$, $SPylZ_j\left[n_{Pylons}\right]$ and $PPylZ_j\left[n_{Pylons}\right]$ are entered monotonically increasing
- Verify that $PWnY_j$ and $PHSY_j$ are entered monotonically decreasing
- Ensure $\left|SWnDhdrl\right| < \dfrac{\pi}{2}$ and $\left|PWnDhdrl\right| < \dfrac{\pi}{2}$
- Check inputs and set parameters

Set the reference positions and orientations of the input line2 meshes:

- Fuselage reference point mesh: $\vec{p}^{\,FusOR} = \vec{0}$; $\Lambda^{FusOR} = I$
- Fuselage: $^{In}\vec{p}_j^{\,FusR} = \begin{Bmatrix} FusX_j \\ FusY_j \\ FusZ_j \end{Bmatrix}$;

  (for $j = \left\{1,2,\ldots,NumFusNds\right\}$) $^{In}\Lambda_j^{FusR} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \Lambda\left(FusTwist_j,0,0\right)$

Commented [JJ8]: Note: the 3x3 matrix can also be written as Lambda(0,90deg,0)

- Starboard (right) wing: $^{In}\vec{p}_j^{\,SWnR} = \vec{p}^{\,SWnOR} + \begin{Bmatrix} SWnX_j \\ SWnY_j \\ SWnZ_j \end{Bmatrix}$;

  (for $j = \left\{1,2,\ldots,NumSWnNds\right\}$) $^{In}\Lambda_j^{SWnR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(-SWnDhdrl, SWnTwist_j, 0\right)$

Commented [JJ9]: Note: the 3x3 matrix can also be written as Lambda(-90deg,0,90deg)

- Port (left) wing: $^{In}\vec{p}_j^{\,PWnR} = \vec{p}^{\,PWnOR} + \begin{Bmatrix} PWnX_j \\ PWnY_j \\ PWnZ_j \end{Bmatrix}$;

  (for $j = \left\{1,2,\ldots,NumPWnNds\right\}$) $^{In}\Lambda_j^{PWnR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(PWnDhdrl, PWnTwist_j, 0\right)$

- Vertical stabilizer:

$$ {}^{In}\vec{p}_j^{VSR} = \vec{p}^{VSOR} + \begin{Bmatrix} VSX_j \\ VSY_j \\ VSZ_j \end{Bmatrix}; $$

(for $j = \{1,2,\dots,NumVSNds\}$)

$$ {}^{In}\Lambda_j^{VSR} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0,0,VSTwist_j\right) $$

Commented [JJ10]: Note: the 3x3 matrix can also be written as Lambda(0,0,90deg)

- Starboard (right) horizontal stabilizer:

$$ {}^{In}\vec{p}_j^{SHSR} = \vec{p}^{SHSOR} + \begin{Bmatrix} SHSX_j \\ SHSY_j \\ SHSZ_j \end{Bmatrix}; $$

(for $j = \{1,2,\dots,NumSHSNds\}$)

$$ {}^{In}\Lambda_j^{SHSR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(0,SHSTwist_j,0\right) $$

Commented [JJ11]: Note: the 3x3 matrix can also be written as Lambda(-90deg,0,90deg)

- Port (left) horizontal stabilizer:

$$ {}^{In}\vec{p}_j^{PHSR} = \vec{p}^{PHSOR} + \begin{Bmatrix} PHSX_j \\ PHSY_j \\ PHSZ_j \end{Bmatrix}; $$

(for $j = \{1,2,\dots,NumPHSNds\}$)

$$ {}^{In}\Lambda_j^{PHSR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(0,PHSTwist_j,0\right) $$

- Starboard (right) pylons :

$$ {}^{In}\vec{p}_j^{SPyR}\left[n_{Pylons}\right] = \vec{p}^{SPyOR}\left[n_{Pylons}\right] + \begin{Bmatrix} SPylX_j\left[n_{Pylons}\right] \\ SPylY_j\left[n_{Pylons}\right] \\ SPylZ_j\left[n_{Pylons}\right] \end{Bmatrix}; $$

(for $j = \{1,2,\dots,NumPylNds\}$)

$$ {}^{In}\Lambda_j^{SPyR}\left[n_{Pylons}\right] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0,0,SPylTwist_j\left[n_{Pylons}\right]\right) $$

Commented [JJ12]: Note: the 3x3 matrix can also be written as Lambda(0,0,90deg)

- Port (left) pylons :

$$ {}^{In}\vec{p}_j^{PPyR}\left[n_{Pylons}\right] = \vec{p}^{PPyOR}\left[n_{Pylons}\right] + \begin{Bmatrix} PPylX_j\left[n_{Pylons}\right] \\ PPylY_j\left[n_{Pylons}\right] \\ PPylZ_j\left[n_{Pylons}\right] \end{Bmatrix}; $$

(for $j = \{1,2,\dots,NumPylNds\}$)

$$ {}^{In}\Lambda_j^{PPyR}\left[n_{Pylons}\right] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0,0,PPylTwist_j\left[n_{Pylons}\right]\right) $$

Set the reference positions and orientations of the output point meshes to be the midpoints of the input meshes (there is one fewer output node than there are input nodes):

- Fuselage:

$$^{Out}\vec{p}_j^{FusR} = \frac{1}{2}\left(\begin{Bmatrix} FusX_j \\ FusY_j \\ FusZ_j \end{Bmatrix} + \begin{Bmatrix} FusX_{j+1} \\ FusY_{j+1} \\ FusZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumFusNds - 1\}$)

$$^{Out}\Lambda_j^{FusR} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \Lambda\left(\frac{1}{2}\left(FusTwist_j + FusTwist_{j+1} \right), 0, 0 \right)$$

- Starboard (right) wing:

$$^{Out}\vec{p}_j^{SWnR} = \vec{p}^{SWnOR} + \frac{1}{2}\left(\begin{Bmatrix} SWnX_j \\ SWnY_j \\ SWnZ_j \end{Bmatrix} + \begin{Bmatrix} SWnX_{j+1} \\ SWnY_{j+1} \\ SWnZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumSWnNds - 1\}$)

$$^{Out}\Lambda_j^{SWnR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(-\frac{1}{2}\left(SWnDhdrl_j + SWnDhdrl_{j+1} \right), \frac{1}{2}\left(SWnTwist_j + SWnTwist_{j+1} \right), 0 \right)$$

- Port (left) wing:

$$^{Out}\vec{p}_j^{PWnR} = \vec{p}^{PWnOR} + \frac{1}{2}\left(\begin{Bmatrix} PWnX_j \\ PWnY_j \\ PWnZ_j \end{Bmatrix} + \begin{Bmatrix} PWnX_{j+1} \\ PWnY_{j+1} \\ PWnZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumPWnNds - 1\}$)

$$^{Out}\Lambda_j^{PWnR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(\frac{1}{2}\left(PWnDhdrl_j + PWnDhdrl_{j+1} \right), \frac{1}{2}\left(PWnTwist_j + PWnTwist_{j+1} \right), 0 \right)$$

- Vertical stabilizer:

$$^{Out}\vec{p}_j^{VSR} = \vec{p}^{VSOR} + \frac{1}{2}\left(\begin{Bmatrix} VSX_j \\ VSY_j \\ VSZ_j \end{Bmatrix} + \begin{Bmatrix} VSX_{j+1} \\ VSY_{j+1} \\ VSZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumVSNds - 1\}$)

$$^{Out}\Lambda_j^{VSR} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0, 0, \frac{1}{2}\left(VSTwist_j + VSTwist_{j+1} \right) \right)$$

- Starboard (right) horizontal stabilizer:

$$^{Out}\vec{p}_j^{SHSR} = \vec{p}^{SHSOR} + \frac{1}{2}\left(\begin{Bmatrix} SHSX_j \\ SHSY_j \\ SHSZ_j \end{Bmatrix} + \begin{Bmatrix} SHSX_{j+1} \\ SHSY_{j+1} \\ SHSZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumSHSNds - 1\}$)

$$^{Out}\Lambda_j^{SHSR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(0, \frac{1}{2}\left(SHSTwist_j + SHSTwist_{j+1} \right), 0 \right)$$

- Port (left) horizontal stabilizer:

$$^{Out}\vec{p}_j^{PHSR} = \vec{p}^{PHSOR} + \frac{1}{2}\left(\begin{Bmatrix} PHSX_j \\ PHSY_j \\ PHSZ_j \end{Bmatrix} + \begin{Bmatrix} PHSX_{j+1} \\ PHSY_{j+1} \\ PHSZ_{j+1} \end{Bmatrix} \right);$$

(for $j = \{1, 2, \ldots, NumPHSNds - 1\}$)

$$^{Out}\Lambda_j^{PHSR} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \Lambda\left(0, \frac{1}{2}\left(PHSTwist_j + PHSTwist_{j+1} \right), 0 \right)$$

- Starboard (right) pylons :

$$ {}^{Out}\vec{p}_j^{SPyR}\left[n_{Pylons}\right] = \vec{p}^{SPyOR}\left[n_{Pylons}\right] + \frac{1}{2}\left(\begin{Bmatrix} SPylX_j\left[n_{Pylons}\right] \\ SPylY_j\left[n_{Pylons}\right] \\ SPylZ_j\left[n_{Pylons}\right] \end{Bmatrix} + \begin{Bmatrix} SPylX_{j+1}\left[n_{Pylons}\right] \\ SPylY_{j+1}\left[n_{Pylons}\right] \\ SPylZ_{j+1}\left[n_{Pylons}\right] \end{Bmatrix} \right); $$

(for $j = \{1, 2, \ldots, NumPylNds - 1\}$)

$$ {}^{Out}\Lambda_j^{SPyR}\left[n_{Pylons}\right] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0, 0, \frac{1}{2}\left(SPylTwist_j\left[n_{Pylons}\right] + SPylTwist_{j+1}\left[n_{Pylons}\right]\right)\right) $$

- Port (left) pylons :

$$ {}^{Out}\vec{p}_j^{PPyR}\left[n_{Pylons}\right] = \vec{p}^{PPyOR}\left[n_{Pylons}\right] + \frac{1}{2}\left(\begin{Bmatrix} PPylX_j\left[n_{Pylons}\right] \\ PPylY_j\left[n_{Pylons}\right] \\ PPylZ_j\left[n_{Pylons}\right] \end{Bmatrix} + \begin{Bmatrix} PPylX_{j+1}\left[n_{Pylons}\right] \\ PPylY_{j+1}\left[n_{Pylons}\right] \\ PPylZ_{j+1}\left[n_{Pylons}\right] \end{Bmatrix} \right); $$

(for $j = \{1, 2, \ldots, NumPylNds - 1\}$)

$$ {}^{Out}\Lambda_j^{PPyR}\left[n_{Pylons}\right] = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Lambda\left(0, 0, \frac{1}{2}\left(PPylTwist_j\left[n_{Pylons}\right] + PPylTwist_{j+1}\left[n_{Pylons}\right]\right)\right) $$

Set the reference positions and orientations of the input/output point meshes for the top and bottom rotors on the pylons :

- on the starboard (right) wing: $\vec{p}^{SPyRtrR}\left[n_{Pylons}, n_2\right] =$ (from initialization input);
  $\Lambda^{SPyRtrR}\left[n_{Pylons}, n_2\right] = I$
- on the port (left) wing: $\vec{p}^{PPyRtrR}\left[n_{Pylons}, n_2\right] =$ (from initialization input);
  $\Lambda^{PPyRtrR}\left[n_{Pylons}, n_2\right] = I$

Call ActuatorDisk_Init()

Set the KiteVSM initialization inputs:

- Chord – Defined per element as the averaged chord between adjacent analysis nodes
- Length – Defined per element as the absolute x distance between adjacent analysis nodes for the fuselage, the absolute y distance between adjacent analysis nodes divided by the COSine of the averaged dihedral angle for the wings (starboard, port), the absolute y distance between adjacent analysis nodes for the horizontal stabilizers (starboard, port), and the absolute z distance for the vertical stabilizer and pylons (starboard, port)
- AirfoilID – Defined per element as the airfoil ID with the smallest x for the fuselage, the smallest y for the starboard wing and horizontal stabilizer, the largest y for the port wing and stabilizer, and the smallest z for the vertical stabilizer and pylons (starboard, port) (instead of interpolating)
- LiftMod – Lifting-line calculation model (-) (switch) {1:geometric AoA, 2:vortex-step method}
- VSMMod – Trailing vortices alignment model (-) (switch) {1:chord, 2: local free stream}
- VSMToler – Tolerance in the Newton iterations (m^2/s) or DEFAULT
- VSMMaxIter – Maximum number of Newton iterations (-) or DEFAULT
- VSMPerturb – Perturbation size for computing the Jacobian in the Newton iterations (m^2/s) or DEFAULT

Call KiteVSM_Init()

Write the KiteAeroDyn summary file if SumPrint = True:
PUT SOMETHING HERE

Open the KiteAeroDyn output file if OutSwtch = 1 or 3.

Update States
Set the KiteVSM inputs at $t + \Delta t$ as follows:

- Global locations of element endpoints (A and B) – set per element:

Commented [JJ13]: I'm assuming KiteVSM also solves the LiftMod=1 case.

Commented [JJ14]: ABS(y_j+1 - y_j)/COS(0.5*(Dihedral_j + Dihedral_j+1))

Commented [JJ15]: We need to set the DEFAULTs.

Commented [JJ16]: We need to set the DEFAULTs.

Commented [JJ17]: We need to set the DEFAULTs.

Commented [jmj18]: Greg has implemented a start of the summary file with the VSM elements, Ax,y,z, Bx,y,z

Commented [JJ19]: Note: KiteVSM does not strictly follow the framework because KiteVSM_UpdateStates() only has one set of inputs at t+dt.

$$\vec{p}_j^{\,A} = {}^{In}\vec{p}_j^{\,FusR} + \vec{u}_j^{\,Fus} \qquad \text{(for } j = \{1,2,\ldots,NumFusNds-1\}\text{)}$$

$$\vec{p}_j^{\,B} = {}^{In}\vec{p}_{j+1}^{\,FusR} + \vec{u}_{j+1}^{\,Fus} \qquad \text{(for } j = \{1,2,\ldots,NumFusNds-1\}\text{)}$$

Commented [JJ20]: KiteVSM needs a way to distinguish between the fuselage and other members because the circulation should be zero on the fuselage.<--The actual implementation separates the fuselage from the other members in some way.

$$\vec{p}_{j+NumFusNds-1}^{\,A} = {}^{In}\vec{p}_j^{\,SWnR} + \vec{u}_j^{\,SWn} \qquad \text{(for } j = \{1,2,\ldots,NumSWnNds-1\}\text{)}$$

$$\vec{p}_{j+NumFusNds-1}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,SWnR} + \vec{u}_{j+1}^{\,SWn} \qquad \text{(for } j = \{1,2,\ldots,NumSWnNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1}}^{\,A} = {}^{In}\vec{p}_j^{\,PWnR} + \vec{u}_j^{\,PWn} \qquad \text{(for } j = \{1,2,\ldots,NumPWnNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1}}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,PWnR} + \vec{u}_{j+1}^{\,PWn} \qquad \text{(for } j = \{1,2,\ldots,NumPWnNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}}^{\,A} = {}^{In}\vec{p}_j^{\,VSR} + \vec{u}_j^{\,VS} \qquad \text{(for } j = \{1,2,\ldots,NumVSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,VSR} + \vec{u}_{j+1}^{\,VS} \qquad \text{(for } j = \{1,2,\ldots,NumVSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1}}^{\,A} = {}^{In}\vec{p}_j^{\,SHSR} + \vec{u}_j^{\,SHS} \qquad \text{(for } j = \{1,2,\ldots,NumSHSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1}}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,SHSR} + \vec{u}_{j+1}^{\,SHS} \qquad \text{(for } j = \{1,2,\ldots,NumSHSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}^{\,A} = {}^{In}\vec{p}_j^{\,PHSR} + \vec{u}_j^{\,PHS} \qquad \text{(for } j = \{1,2,\ldots,NumPHSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,PHSR} + \vec{u}_{j+1}^{\,PHS} \qquad \text{(for } j = \{1,2,\ldots,NumPHSNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)(n_{Pylons}-1)}}^{\,A} = {}^{In}\vec{p}_j^{\,SPyR}\left[n_{Pylons}\right] + \vec{u}_j^{\,SPy}\left[n_{Pylons}\right] \qquad \text{(for } j = \{1,2,\ldots,NumPylNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)(n_{Pylons}-1)}}^{\,B} = {}^{In}\vec{p}_{j+1}^{\,SPyR}\left[n_{Pylons}\right] + \vec{u}_{j+1}^{\,SPy}\left[n_{Pylons}\right] \qquad \text{(for } j = \{1,2,\ldots,NumPylNds-1\}\text{)}$$

$$\vec{p}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)N_{Pylons}\\+(NumPylNds-1)(n_{Pylons}-1)}}^{\,A} = {}^{In}\vec{p}_j^{\,PPyR}\left[n_{Pylons}\right] + \vec{u}_j^{\,PPy}\left[n_{Pylons}\right] \qquad \text{(for } j = \{1,2,\ldots,NumPylNds-1\}\text{)}$$

$$\vec{p}^{\,B}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)N_{Pylons}\\+(NumPylNds-1)(n_{Pylons}-1)}} = {}^{In}\vec{p}^{\,PPyR}_{j+1}\left[n_{Pylons}\right] + \vec{u}^{\,PPy}_{j+1}\left[n_{Pylons}\right] \quad \text{(for } j=\{1,2,\ldots,NumPylNds-1\}\text{)}$$

- Velocities – set per element:

$$ {}^{Rel}\vec{V}_j = \frac{1}{2}\left(\vec{V}^{Fus}_j + \vec{V}^{Fus}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{Fus}_j + \vec{v}^{Fus}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumFusNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{j+NumFusNds-1} = \frac{1}{2}\left(\vec{V}^{SWn}_j + \vec{V}^{SWn}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{SWn}_j + \vec{v}^{SWn}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumSWnNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1}} = \frac{1}{2}\left(\vec{V}^{PWn}_j + \vec{V}^{PWn}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{PWn}_j + \vec{v}^{PWn}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumPWnNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}} = \frac{1}{2}\left(\vec{V}^{VS}_j + \vec{V}^{VS}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{VS}_j + \vec{v}^{VS}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumVSNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1}} = \frac{1}{2}\left(\vec{V}^{SHS}_j + \vec{V}^{SHS}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{SHS}_j + \vec{v}^{SHS}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumSHSNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}} = \frac{1}{2}\left(\vec{V}^{PHS}_j + \vec{V}^{PHS}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{PHS}_j + \vec{v}^{PHS}_{j+1}\right) \quad \text{(for } j=\{1,2,\ldots,NumPHSNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)(n_{Pylons}-1)}} = \left\{\frac{1}{2}\left(\vec{V}^{SPy}_j + \vec{V}^{SPy}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{SPy}_j + \vec{v}^{SPy}_{j+1}\right)\right\}\left[n_{Pylons}\right] \quad \text{(for } j=\{1,2,\ldots,NumPylNds-1\}\text{)}$$

$$ {}^{Rel}\vec{V}_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)N_{Pylons}\\+(NumPylNds-1)(n_{Pylons}-1)}} = \left\{\frac{1}{2}\left(\vec{V}^{PPy}_j + \vec{V}^{PPy}_{j+1}\right) - \frac{1}{2}\left(\vec{v}^{PPy}_j + \vec{v}^{PPy}_{j+1}\right)\right\}\left[n_{Pylons}\right] \quad \text{(for } j=\{1,2,\ldots,NumPylNds-1\}\text{)}$$

- Aerodynamic + elastic twist – set per element:

Commented [JJ21]: To do these Line2-to-Point mesh mappings, the orientation field can be added to a MiscVar that is a Sibling of the output mesh.

$$ {}^{Out}\Lambda_j = M^{L2P}_{\Lambda}\left(\Lambda^{Fus}\right) \quad \text{(for } j=\{1,2,\ldots,NumFusNds-1\}\text{)}$$

$$ {}^{Out}\Lambda_{j+NumFusNds-1} = M^{L2P}_{\Lambda}\left(\Lambda^{SWn}\right) \quad \text{(for } j=\{1,2,\ldots,NumSWnNds-1\}\text{)}$$

$$ {}^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1}} = M^{L2P}_{\Lambda}\left(\Lambda^{PWn}\right) \quad \text{(for } j=\{1,2,\ldots,NumPWnNds-1\}\text{)}$$

$$ {}^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}} = M^{L2P}_{\Lambda}\left(\Lambda^{VS}\right) \quad \text{(for } j=\{1,2,\ldots,NumVSNds-1\}\text{)}$$

$$ {}^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1}} = M^{L2P}_{\Lambda}\left(\Lambda^{SHS}\right) \quad \text{(for } j=\{1,2,\ldots,NumSHSNds-1\}\text{)}$$

$$^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}} = M_{\Lambda}^{L2P}\left(\Lambda^{PHS}\right) \qquad \text{(for } j=\left\{1,2,\ldots,NumPHSNds-1\right\}\text{)}$$

$$^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)(n_{Pylons}-1)}} = M_{\Lambda}^{L2P}\left(\Lambda^{SPy}\left[n_{Pylons}\right]\right) \qquad \text{(for } j=\left\{1,2,\ldots,NumPylNds-1\right\}\text{)}$$

$$^{Out}\Lambda_{\substack{j+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1\\+NumPHSNds-1\\+(NumPylNds-1)N_{Pylons}\\+(NumPylNds-1)(n_{Pylons}-1)}} = M_{\Lambda}^{L2P}\left(\Lambda^{PPy}\left[n_{Pylons}\right]\right) \qquad \text{(for } j=\left\{1,2,\ldots,NumPylNds-1\right\}\text{)}$$

- Control settings – Defined per element as the control setting with the smallest y for the starboard wing and horizontal stabilizer, the largest y for the port wing and stabilizer, and the smallest z for the vertical stabilizer (instead of interpolating)

Call KiteVSM_UpdateStates()

No need to call ActuatorDisk_UpdateStates(), as this is blank anyway…

Calculate Output

Calculate the output-mesh motions of the fuselage, wings, vertical stabilizer, horizontal stabilizers, and pylons via mesh mapping:

$$^{Out}\vec{u}_j^{Fus} = M_u^{L2P}\left(\vec{u}_j^{Fus}, \Lambda_j^{Fus}\right)$$

$$^{Out}\Lambda_j^{Fus} = M_{\Lambda}^{L2P}\left(\Lambda_j^{Fus}\right)$$

$$^{Out}\vec{u}_j^{SWn} = M_u^{L2P}\left(\vec{u}_j^{SWn}, \Lambda_j^{SWn}\right)$$

$$^{Out}\Lambda_j^{SWn} = M_{\Lambda}^{L2P}\left(\Lambda_j^{SWn}\right)$$

$$^{Out}\vec{u}_j^{PWn} = M_u^{L2P}\left(\vec{u}_j^{PWn}, \Lambda_j^{PWn}\right)$$

$$^{Out}\Lambda_j^{PWn} = M_{\Lambda}^{L2P}\left(\Lambda_j^{PWn}\right)$$

$$^{Out}\vec{u}_j^{VS} = M_u^{L2P}\left(\vec{u}_j^{VS}, \Lambda_j^{VS}\right)$$

$$^{Out}\Lambda_j^{VS} = M_{\Lambda}^{L2P}\left(\Lambda_j^{VS}\right)$$

$$^{Out}\vec{u}_j^{SHS} = M_u^{L2P}\left(\vec{u}_j^{SHS}, \Lambda_j^{SHS}\right)$$

$$^{Out}\Lambda_j^{SHS} = M_{\Lambda}^{L2P}\left(\Lambda_j^{SHS}\right)$$

$$^{Out}\vec{u}_j^{PHS} = M_u^{L2P}\left(\vec{u}_j^{PHS}, \Lambda_j^{PHS}\right)$$

$$^{Out}\Lambda_j^{PHS} = M_{\Lambda}^{L2P}\left(\Lambda_j^{PHS}\right)$$

$$^{Out}\vec{u}_j^{SPy}\left[n_{Pylons}\right] = M_u^{L2P}\left(\vec{u}_j^{SPy}\left[n_{Pylons}\right], \Lambda_j^{SPy}\left[n_{Pylons}\right]\right)$$

$$^{Out}\Lambda_j^{SPy}\left[n_{Pylons}\right] = M_{\Lambda}^{L2P}\left(\Lambda_j^{SPy}\left[n_{Pylons}\right]\right)$$

$$ {}^{Out}\vec{u}_j^{PPy}\left[n_{Pylons}\right] = M_u^{L2P}\left(\vec{u}_j^{PPy}\left[n_{Pylons}\right], \Lambda_j^{PPy}\left[n_{Pylons}\right]\right) $$
$$ {}^{Out}\Lambda_j^{PPy}\left[n_{Pylons}\right] = M_\Lambda^{L2P}\left(\Lambda_j^{PPy}\left[n_{Pylons}\right]\right) $$

Calculate the aerodynamic loads on the fuselage, wings, vertical stabilizer, horizontal stabilizers, and pylons as follows:

- Set the KiteVSM inputs as in Update States above
- Call KiteVSM_CalcOutput()
- Transfer the aerodynamic loads—per element—calculated by KiteVSM to the KiteAeroDyn output mesh:

$\vec{F}_j^{Fus} = \vec{F}_j$ and $\vec{M}_j^{Fus} = \vec{M}_j$ (for $j = \{1, 2, \ldots, NumFusNds - 1\}$)

$\vec{F}_j^{SWn} = \vec{F}_{j+NumFusNds-1}$ and $\vec{M}_j^{SWn} = \vec{M}_{j+NumFusNds-1}$ (for $j = \{1, 2, \ldots, NumSWnNds - 1\}$)

$\vec{F}_j^{PWn} = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1}}$ and $\vec{M}_j^{PWn} = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1}}$ (for $j = \{1, 2, \ldots, NumPWnNds - 1\}$)

$\vec{F}_j^{VS} = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}$ and $\vec{M}_j^{VS} = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}$ (for $j = \{1, 2, \ldots, NumVSNds - 1\}$)

$\vec{F}_j^{SHS} = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}$ and $\vec{M}_j^{SHS} = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}$ (for $j = \{1, 2, \ldots, NumSHSNds - 1\}$)

$\vec{F}_j^{PHS} = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}$ and $\vec{M}_j^{PHS} = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}$ (for $j = \{1, 2, \ldots, NumPHSNds - 1\}$)

$\vec{F}_j^{SPy}\left[n_{Pylons}\right] = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(n_{Pylons}-1)}}$ and $\vec{M}_j^{SPy}\left[n_{Pylons}\right] = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(n_{Pylons}-1)}}$ (for $j = \{1, 2, \ldots, NumPylNds - 1\}$)

$\vec{F}_j^{PPy}\left[n_{Pylons}\right] = \vec{F}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(n_{Pylons}-1)}}$ and $\vec{M}_j^{PPy}\left[n_{Pylons}\right] = \vec{M}_{\substack{j+NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(n_{Pylons}-1)}}$ (for $j = \{1, 2, \ldots, NumPylNds - 1\}$)

Calculate the aerodynamic rotor loads as follows:

IF $\left(RotorMod = 1\right)$ THEN

The calculations below are in a loop for $n_{Pylons} = \{1, 2, \ldots, N_{Pylons}\}$ and $n_2 = \{1, 2\}$. Also the calculations for the rotors on the starboard wing should be repeated for the rotors on the port wing.

Set inputs to ActuatorDisk:

- $\Omega = \Omega^{SPyRtr}\left[n_{Pylons}, n_2\right]$
- $\theta = \theta^{SPyRtr}\left[n_{Pylons}, n_2\right]$
- $V^{Rel} = \left\| {}^{Rel}\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right] \right\|_2$

- $$\chi = \begin{cases} 0 & for\left(V^{Rel} = 0\right) \\ ACOS\left(\dfrac{ {}^{Rel}\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right] \bullet \hat{x}^{SPyRtr}\left[n_{Pylons}, n_2\right] }{ V^{Rel} } \right) & otherwise \end{cases}$$

  where:
  $${}^{Rel}\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right] = \vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right] - \vec{v}^{SPyRtr}\left[n_{Pylons}, n_2\right]$$

Call ActuatorDisk_CalcOutput()

Set KiteAeroDyn outputs related to the rotors:

- $$\vec{F}^{SPyRtr}\left[n_{Pylons}, n_2\right] = \left[\Lambda^{Disk}\left[n_{Pylons}, n_2\right]\right]^T \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix}$$
- $$\vec{M}^{SPyRtr}\left[n_{Pylons}, n_2\right] = \left[\Lambda^{Disk}\left[n_{Pylons}, n_2\right]\right]^T \begin{Bmatrix} M_x \\ M_y \\ M_z \end{Bmatrix}$$

  where a local disk coordinate system is defined such that $\hat{x}^{Disk}\left[n_{Pylons}, n_2\right]$ is normal to the disk, $\hat{y}^{Disk}\left[n_{Pylons}, n_2\right] / \hat{z}^{Disk}\left[n_{Pylons}, n_2\right]$ are in the disk, ${}^{Rel}\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right]$ is in the $\hat{x}^{Disk}\left[n_{Pylons}, n_2\right] / \hat{y}^{Disk}\left[n_{Pylons}, n_2\right]$ plane with positive ${}^{Rel}\vec{V}^{SPyRtr}\left[n_{Pylons}, n_2\right]$ along negative $\hat{y}^{Disk}\left[n_{Pylons}, n_2\right]$, and $\hat{z}^{Disk}\left[n_{Pylons}, n_2\right]$ is normal to this plane as follows:

  $$\begin{Bmatrix} \hat{x}^{Disk} \\ \hat{y}^{Disk} \\ \hat{z}^{Disk} \end{Bmatrix}\left[n_{Pylons}, n_2\right] = \begin{cases} \begin{Bmatrix} \hat{x}^{SPyRtr} \\ \hat{y}^{SPyRtr} \\ \hat{z}^{SPyRtr} \end{Bmatrix}\left[n_{Pylons}, n_2\right] & for\left(\left\| \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{RPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} \right\|_2 \left[n_{Pylons}, n_2\right] = 0 \right) \\ \begin{Bmatrix} \hat{x}^{SPyRtr} \\ \dfrac{ \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{SPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} }{ \left\| \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{SPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} \right\|_2 } \\ \hat{x}^{SPyRtr} \times \hat{y}^{Disk} \end{Bmatrix}\left[n_{Pylons}, n_2\right] = \begin{Bmatrix} \hat{x}^{SPyRtr} \\ \dfrac{ \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{SPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} }{ \left\| \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{SPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} \right\|_2 } \\ \dfrac{ {}^{Rel}\vec{V}^{SPyRtr} \times \hat{x}^{SPyRtr} }{ \left\| \left({}^{Rel}\vec{V}^{SPyRtr} \bullet \hat{x}^{SPyRtr} \right)\hat{x}^{SPyRtr} - {}^{Rel}\vec{V}^{SPyRtr} \right\|_2 } \end{Bmatrix}\left[n_{Pylons}, n_2\right] & otherwise \end{cases}$$

ELSE

The calculations below are in a loop for $n_{Pylons} = \left\{1, 2, \ldots, N_{Pylons}\right\}$ and $n_2 = \left\{1, 2\right\}$. Also the calculations for the rotors on the starboard wing should be repeated for the rotors on the port wing.

Set KiteAeroDyn outputs related to the rotors:

- $\vec{F}^{SPyRtr}\left[n_{Pylons}, n_2\right] = \vec{0}$
- $\vec{M}^{SPyRtr}\left[n_{Pylons}, n_2\right] = \vec{0}$

END IF

Calculate the write outputs and write outputs to a file as follows:

This is a list of all possible output parameters available within the KiteAeroDyn module. The names are grouped by meaning, but can be ordered in the OUTPUTS section of the KiteAeroDyn input file as you see fit.

Fusβ refers to output node β on the fuselage, where β is a one-digit number in the range [1,9] corresponding to the center of the element where entry β in the ***FusOutNd*** list defines the endpoint with the smallest x. Setting β > ***NFusOuts*** yields invalid output.

Commented [JJ22]: Because the outputs are based on the centers of the elements, and not the element end points, the FusOutNd list cannot contain node NumFusNds. Likewise for the wings, stabilizers, and pylons.

SWnβ and PWnβ refer to output node β on the starboard and port wings, respectively, where β is a one-digit number in the range [1,9] corresponding to the center of the element where entry β in the ***SWnOutNd*** and ***PWnOutNd*** lists define the endpoints with the smallest y and largest y, respectively. Setting β > ***NSWnOuts*** and ***NPWnOuts***, respectively, yields invalid output. SFlpα and PFlpα refer to flap α on the starboard and port wings, respectively, where α is a one-digit number in the range [1,9]. If ***NumFlaps*** > 9, only the first 9 flaps can be output.

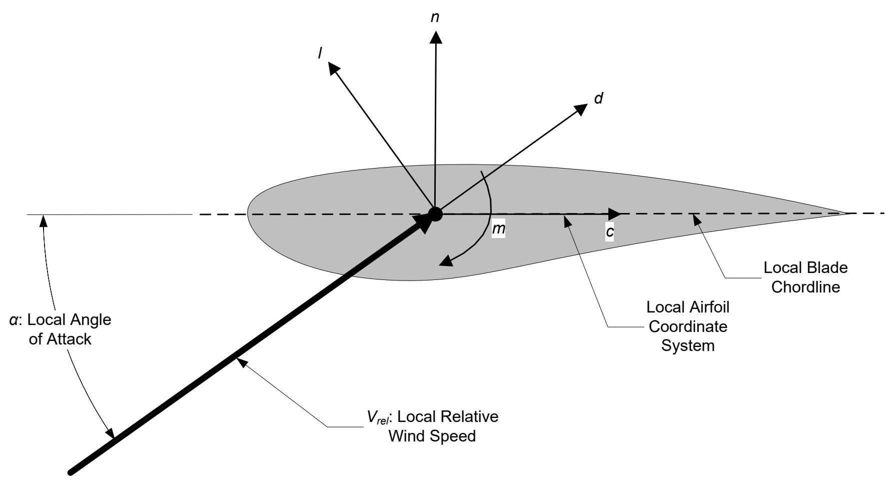
VSβ refers to output node β on the vertical stabilizer, where β is a one-digit number in the range [1,9] corresponding to the center of the element where entry β in the ***VSOutNd*** list defines the endpoint with the smallest z. Setting β > ***NVSOuts*** yields invalid output. Rudrα refers to rudder α on the vertical stabilizer, where α is a one-digit number in the range [1,2].

SHSβ and PHSβ refer to output node β on the starboard and port horizontal stabilizers, respectively, where β is a one-digit number in the range [1,9] corresponding to the center of the element where entry β in the ***SHSOutNd*** and ***PHSOutNd*** lists define the endpoints with the smallest y and largest y, respectively. Setting β > ***NSHSOuts*** and ***NPHSOuts***, respectively, yields invalid output. SElvα and PElvα refer to elevator α on the starboard and port horizontal stabilizers, respectively, where α is a one-digit number in the range [1,2].

SPα and PPα refer to pylon α on the starboard and port wings, respectively, where α is a one-digit number in the range [1,9]. SPαβ and PPαβ refer to output node β on pylon α on the starboard and port wings, respectively, where α is a one-digit number in the range [1,9] and β is a one-digit number in the range [1,9] corresponding to the center of the element where entry β in the ***PylOutNd*** list defines the endpoint with the smallest z. Setting α > ***NumPylons*** or setting β > ***NPylOuts*** yields invalid output. If ***NumPylons*** > 9, only the first 9 pylons can be output.

For the fuselage, wings, vertical stabilizer, horizontal stabilizers, and pylons, the local airfoil coordinate system, including the local angle of attack and force components, is shown below. The spanwise (s) axis is not shown, but is directed into the page following the right-hand rule i.e. $s = n \times c$, where n is normal to the chord pointed toward the suction surface and c is along the chord pointed toward the trailing edge.

Commented [JJ23]: The local coordinate system in KiteVSM has x=n, y=c, and z=s.



Channel Name(s)	Unit(s)	Description
Fuselage		
FusβVAmbn, FusβVAmbc, FusβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at Fusβ in the local airfoil coordinate system
FusβSTVn, FusβSTVc, FusβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at Fusβ in the local airfoil coordinate system
FusβVRel	(m/s)	Relative wind speed at Fusβ
FusβDynP	(Pa)	Dynamic pressure at Fusβ
FusβRe, FusβM	(-), (-)	Reynolds number (in millions) and Mach number at Fusβ
FusβVIndn, FusβVIndc, FusβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at Fusβ in the local airfoil coordinate system
FusβAlpha	(deg)	Angle of attack at Fusβ
FusβCl, FusβCd, FusβCm, FusβCn, FusβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at Fusβ
FusβFl, FusβFd, FusβMm, FusβFn, FusβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at Fusβ
Starboard (Right) Wing		
SWnβVAmbn, SWnβVAmbc, SWnβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at SWnβ in the local airfoil coordinate system
SWnβSTVn, SWnβSTVc, SWnβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at SWnβ in the local airfoil coordinate system
SWnβVRel	(m/s)	Relative wind speed at SWnβ
SWnβDynP	(Pa)	Dynamic pressure at SWnβ
SWnβRe, SWnβM	(-), (-)	Reynolds number (in millions) and Mach number at SWnβ
SWnβVIndn, SWnβVIndc, SWnβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at SWnβ in the local airfoil coordinate system
SWnβAlpha	(deg)	Angle of attack at SWnβ
SFlpαCtrl	(user)	Control setting of flap SFlpα

SWnβCl, SWnβCd, SWnβCm, SWnβCn, SWnβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at SWnβ
SWnβFl, SWnβFd, SWnβMm, SWnβFn, SWnβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at SWnβ
Port (Left) Wing		
PWnβVAmbn, PWnβVAmbc, PWnβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at PWnβ in the local airfoil coordinate system
PWnβSTVn, PWnβSTVc, PWnβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at PWnβ in the local airfoil coordinate system
PWnβVRel	(m/s)	Relative wind speed at PWnβ
PWnβDynP	(Pa)	Dynamic pressure at PWnβ
PWnβRe, PWnβM	(-), (-)	Reynolds number (in millions) and Mach number at PWnβ
PWnβVIndn, PWnβVIndc, PWnβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at PWnβ in the local airfoil coordinate system
PWnβAlpha	(deg)	Angle of attack at PWnβ
PFlpαCtrl	(user)	Control setting of flap PFlpα
PWnβCl, PWnβCd, PWnβCm, PWnβCn, PWnβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at PWnβ
PWnβFl, PWnβFd, PWnβMm, PWnβFn, PWnβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at PWnβ
Vertical Stabilizer		
VSβVAmbn, VSβVAmbc, VSβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at VSβ in the local airfoil coordinate system
VSβSTVn, VSβSTVc, VSβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at VSβ in the local airfoil coordinate system
VSβVRel	(m/s)	Relative wind speed at VSβ
VSβDynP	(Pa)	Dynamic pressure at VSβ
VSβRe, VSβM	(-), (-)	Reynolds number (in millions) and Mach number at VSβ
VSβVIndn, VSβVIndc, VSβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at VSβ in the local airfoil coordinate system
VSβAlpha	(deg)	Angle of attack at VSβ
RudrαCtrl	(user)	Control setting of rudder Rudrα
VSβCl, VSβCd, VSβCm, VSβCn, VSβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at VSβ
VSβFl, VSβFd, VSβMm, VSβFn, VSβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at VSβ
Starboard (Right) Horizontal Stabilizer		
SHSβVAmbn, SHSβVAmbc, SHSβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at SHSβ in the local airfoil coordinate system
SHSβSTVn, SHSβSTVc, SHSβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at SHSβ in the local airfoil coordinate system
SHSβVRel	(m/s)	Relative wind speed at SHSβ
SHSβDynP	(Pa)	Dynamic pressure at SHSβ
SHSβRe, SHSβM	(-), (-)	Reynolds number (in millions) and Mach number at SHSβ
SHSβVIndn, SHSβVIndc, SHSβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at SHSβ in the local airfoil

		coordinate system
SHSβAlpha	(deg)	Angle of attack at SHSβ
SElvαCtrl	(user)	Control setting of elevator SElvα
SHSβCl, SHSβCd, SHSβCm, SHSβCn, SHSβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at SHSβ
SHSβFl, SHSβFd, SHSβMm, SHSβFn, SHSβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at SHSβ
Port (Left) Horizontal Stabilizer		
PHSβVAmbn, PHSβVAmbc, PHSβVAmbs	(m/s), (m/s), (m/s)	Ambient wind velocity at PHSβ in the local airfoil coordinate system
PHSβSTVn, PHSβSTVc, PHSβSTVs	(m/s), (m/s), (m/s)	Structural translational velocity at PHSβ in the local airfoil coordinate system
PHSβVRel	(m/s)	Relative wind speed at PHSβ
PHSβDynP	(Pa)	Dynamic pressure at PHSβ
PHSβRe, PHSβM	(-), (-)	Reynolds number (in millions) and Mach number at PHSβ
PHSβVIndn, PHSβVIndc, PHSβVInds	(m/s), (m/s), (m/s)	Induced wind velocity at PHSβ in the local airfoil coordinate system
PHSβAlpha	(deg)	Angle of attack at PHSβ
PElvαCtrl	(user)	Control setting of elevator PElvα
PHSβCl, PHSβCd, PHSβCm, PHSβCn, PHSβCc	(-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at PHSβ
PHSβFl, PHSβFd, PHSβMm, PHSβFn, PHSβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at PHSβ
Pylons		
SPαβVAmbn, SPαβVAmbc, SPαβVAmbs, PPαβVAmbn, PPαβVAmbc, PPαβVAmbs	(m/s), (m/s), (m/s), (m/s), (m/s), (m/s)	Ambient wind velocity at SPαβ and PPαβ in the local airfoil coordinate system
SPαβSTVn, SPαβSTVc, SPαβSTVs, PPαβSTVn, PPαβSTVc, PPαβSTVs	(m/s), (m/s), (m/s), (m/s), (m/s), (m/s)	Structural translational velocity at SPαβ and PPαβ in the local airfoil coordinate system
SPαβVRel, PPαβVRel	(m/s), (m/s)	Relative wind speed at SPαβ and PPαβ
SPαβDynP, PPαβDynP	(Pa), (Pa)	Dynamic pressure at SPαβ and PPαβ
SPαβRe, SPαβM, PPαβRe, PPαβM	(-), (-), (-), (-)	Reynolds number (in millions) and Mach number at SPαβ and PPαβ
SPαβVIndn, SPαβVIndc, SPαβVInds PPαβVIndn, PPαβVIndc, PPαβVInds	(m/s), (m/s), (m/s), (m/s), (m/s), (m/s)	Induced wind velocity at SPαβ and PPαβ in the local airfoil coordinate system
SPαβAlpha, PPαβAlpha	(deg), (deg)	Angle of attack at SPαβ and PPαβ
SPαβCl, SPαβCd, SPαβCm, SPαβCn, SPαβCc, PPαβCl, PPαβCd, PPαβCm, PPαβCn, PPαβCc	(-), (-), (-), (-), (-), (-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force coefficients at SPαβ and PPαβ
SPαβFl, SPαβFd, SPαβMm, SPαβFn, SPαβFc, PPαβFl, PPαβFd, PPαβMm, PPαβFn, PPαβFc	(N/m), (N/m), (N·m/m), (N/m), (N/m), (N/m), (N/m), (N·m/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to chord), and chordwise force per unit length at SPαβ and PPαβ
Rotors		
SPαTTSR, SPαBTSR, PPαTTSR, PPαBTSR	(-), (-), (-), (-)	Rotor tip-speed ratio of the top (T) and bottom (B) rotor on SPα and PPα
SPαTPitch, SPαBPitch,	(deg), (deg),	Rotor-collective blade-pitch angle of the top (T)

PPαTPitch, PPαBPitch	(deg), (deg)	and bottom (B) rotor on SPα and PPα
SPαTSkew, SPαBSkew, PPαTSkew, PPαBSkew	(deg), (deg), (deg), (deg)	Rotor inflow-skew angle of the top (T) and bottom (B) rotor on SPα and PPα
SPαTRtSpd, SPαBRtSpd, PPαTRtSpd, PPαBRtSpd	(rad/s), (rad/s), (rad/s), (rad/s)	Rotor speed of the top (T) and bottom (B) rotor on SPα and PPα
SPαTVRel, SPαBVRel, PPαTVRel, PPαBVRel	(m/s), (m/s), (m/s), (m/s)	Rotor-disk-averaged relative wind speed of the top (T) and bottom (B) rotor on SPα and PPα
SPαTCp, SPαBCp, PPαTCp, PPαBCp, SPαTCq, SPαBCq, PPαTCq, PPαBCq, SPαTCt, SPαBCt, PPαTCt, PPαBCt	(-), (-), (-), (-), (-), (-), (-), (-), (-), (-), (-), (-)	Rotor power, torque, and thrust coefficients of the top (T) and bottom (B) rotor on SPα and PPα
SPαTFx, SPαBFx, PPαTFx, PPαBFx, SPαTFy, SPαBFy, PPαTFy, PPαBFy, SPαTFz, SPαBFz, PPαTFz, PPαBFz, SPαTMx, SPαBMx, PPαTMx, PPαBMx, SPαTMy, SPαBMy, PPαTMy, PPαBMy, SPαTMz, SPαBMz, PPαTMz, PPαBMz	(N), (N), (N), (N), (N), (N), (N), (N), (N), (N), (N), (N), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m), (N·m)	Rotor aerodynamic loads in the local coordinate system of the top (T) and bottom (B) rotor on SPα and PPα
SPαTPwr, SPαBPwr, PPαTPwr, PPαBPwr	(W), (W), (W), (W)	Rotor power of the top (T) and bottom (B) rotor on SPα and PPα
Energy Kite		
KiteFxi, KiteFyi, KiteFzi, KiteMxi, KiteMyi, KiteMzi	(N), (N), (N), (N·m), (N·m), (N·m)	Total integrated aerodynamic loads applied to the energy kite about the fuselage reference point in the global inertial-frame coordinate system
KitePwr	(W)	Total power from all rotors

Commented [JJ24]: This coordinate system is coincident with the body-fixed local coordinate of the energy kite, but following the deflections of the wings and pylons. The coordinate system is not associated with the azimuth or inflow-skew angle of the rotor.

These are calculated within KiteAeroDyn as follows:

Fuselage:

$$\begin{Bmatrix} Fus\beta VAmbn \\ Fus\beta VAmbc \\ Fus\beta VAmbs \end{Bmatrix} = {}^{Out}\Lambda^{Fus}_{FusOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{V}^{Fus}_{FusOutNd[\beta]} + \vec{V}^{Fus}_{FusOutNd[\beta]+1} \right) \right\}$$

$$\begin{Bmatrix} Fus\beta STVn \\ Fus\beta STVc \\ Fus\beta STVs \end{Bmatrix} = {}^{Out}\Lambda^{Fus}_{FusOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{v}^{Fus}_{FusOutNd[\beta]} + \vec{v}^{Fus}_{FusOutNd[\beta]+1} \right) \right\}$$

$$Fus\beta VRel = U_{rel_{FusOutNd[\beta]}}$$

Commented [JJ25]: U_rel is from the VSM theory documentation; likewise below.

$$Fus\beta DynP = \frac{1}{2}\rho U^2_{rel_{FusOutNd[\beta]}}$$

$$\begin{Bmatrix} Fus\beta\,Re \\ Fus\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{FusOutNd[\beta]} U_{rel_{FusOutNd[\beta]}}}{1000000\,KinVisc} \\ \dfrac{U_{rel_{FusOutNd[\beta]}}}{SpdSound} \end{Bmatrix}$$

Commented [JJ26]: c is from the VSM theory documentation; likewise below.

$$\begin{Bmatrix} Fus\beta VIndn \\ Fus\beta VIndc \\ Fus\beta VInds \end{Bmatrix} = {}^{Out}\Lambda^{Fus}_{FusOutNd[\beta]} \vec{U}_{ind_{FusOutNd[\beta]}}$$

Commented [JJ27]: U_ind is from the VSM theory documentation; likewise below.

$$Fus\beta Alpha = \frac{180}{\pi}\alpha_{FusOutNd[\beta]}$$

Commented [JJ28]: alpha is from the VSM theory documentation; likewise below.

$$\begin{Bmatrix} Fus\beta Cl \\ Fus\beta Cd \\ Fus\beta Cm \\ Fus\beta Cn \\ Fus\beta Cc \end{Bmatrix} = \begin{Bmatrix} C_{l_{FusOutNd[\beta]}} \\ C_{d_{FusOutNd[\beta]}} \\ C_{m_{FusOutNd[\beta]}} \\ C_{l_{FusOutNd[\beta]}} COS\left(\alpha_{FusOutNd[\beta]}\right) + C_{d_{FusOutNd[\beta]}} SIN\left(\alpha_{FusOutNd[\beta]}\right) \\ -C_{l_{FusOutNd[\beta]}} SIN\left(\alpha_{FusOutNd[\beta]}\right) + C_{d_{FusOutNd[\beta]}} COS\left(\alpha_{FusOutNd[\beta]}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} Fus\beta Fl \\ Fus\beta Fd \\ Fus\beta Mm \\ Fus\beta Fn \\ Fus\beta Fc \end{Bmatrix} = \begin{Bmatrix} (Fus\beta Dyn)(Fus\beta Cl) \\ (Fus\beta Dyn)(Fus\beta Cd) \\ (Fus\beta Dyn)c_{FusOutNd[\beta]}(Fus\beta Cm) \\ (Fus\beta Dyn)(Fus\beta Cn) \\ (Fus\beta Dyn)(Fus\beta Cc) \end{Bmatrix}$$

Commented [JJ29]: C_l, C_d, and C_m are from the VSM theory documentation; likewise below.

Starboard (Right) Wing

$$\begin{Bmatrix} SWn\beta VAmbn \\ SWn\beta VAmbc \\ SWn\beta VAmbs \end{Bmatrix} = {}^{Out}\Lambda^{SWn}_{SWnOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{V}^{SWn}_{SWnOutNd[\beta]} + \vec{V}^{SWn}_{SWnOutNd[\beta]+1}\right) \right\}$$

$$\begin{Bmatrix} SWn\beta STVn \\ SWn\beta STVc \\ SWn\beta STVs \end{Bmatrix} = {}^{Out}\Lambda^{SWn}_{SWnOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{v}^{SWn}_{SWnOutNd[\beta]} + \vec{v}^{SWn}_{SWnOutNd[\beta]+1}\right) \right\}$$

$$SWn\beta VRel = U_{rel_{SWnOutNd[\beta] + NumFusNds-1}}$$

$$SWn\beta DynP = \frac{1}{2}\rho U^2_{rel_{SWnOutNd[\beta] + NumFusNds-1}}$$

$$\begin{Bmatrix} SWn\beta Re \\ SWn\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{SWnOutNd[\beta] + NumFusNds-1}\, U_{rel_{SWnOutNd[\beta] + NumFusNds-1}}}{1000000 KinVisc} \\ \dfrac{U_{rel_{SWnOutNd[\beta] + NumFusNds-1}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} SWn\beta VIndn \\ SWn\beta VIndc \\ SWn\beta VInds \end{Bmatrix} = {}^{Out}\Lambda^{SWn}_{SWnOutNd[\beta]} \vec{U}_{ind_{SWnOutNd[\beta] + NumFusNds-1}}$$

$$SWn\beta Alpha = \frac{180}{\pi}\alpha_{SWnOutNd[\beta] + NumFusNds-1}$$

$$SFlp\alpha Ctrl = Ctrl^{SFlp}[\alpha]$$

Commented [JJ30]: This must map the correct output node to the correct control ID. Likewise below.

$$\begin{Bmatrix} SWn\beta Cl \\ SWn\beta Cd \\ SWn\beta Cm \\ SWn\beta Cn \\ SWn\beta Cc \end{Bmatrix} = \begin{Bmatrix} C_{l_{SWnOutNd[\beta]+NumFusNds-1}} \\ C_{d_{SWnOutNd[\beta]+NumFusNds-1}} \\ C_{m_{SWnOutNd[\beta]+NumFusNds-1}} \\ C_{l_{SWnOutNd[\beta]+NumFusNds-1}} COS\left(\alpha_{SWnOutNd[\beta]+NumFusNds-1}\right) + C_{d_{SWnOutNd[\beta]+NumFusNds-1}} SIN\left(\alpha_{SWnOutNd[\beta]+NumFusNds-1}\right) \\ -C_{l_{SWnOutNd[\beta]+NumFusNds-1}} SIN\left(\alpha_{SWnOutNd[\beta]+NumFusNds-1}\right) + C_{d_{SWnOutNd[\beta]+NumFusNds-1}} COS\left(\alpha_{SWnOutNd[\beta]+NumFusNds-1}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} SWn\beta Fl \\ SWn\beta Fd \\ SWn\beta Mm \\ SWn\beta Fn \\ SWn\beta Fc \end{Bmatrix} = \begin{Bmatrix} (SWn\beta Dyn)(SWn\beta Cl) \\ (SWn\beta Dyn)(SWn\beta Cd) \\ (SWn\beta Dyn)c_{SWnOutNd[\beta]+NumFusNds-1}(SWn\beta Cm) \\ (SWn\beta Dyn)(SWn\beta Cn) \\ (SWn\beta Dyn)(SWn\beta Cc) \end{Bmatrix}$$

Port (Left) Wing

$$\begin{Bmatrix} PWn\beta VAmbn \\ PWn\beta VAmbc \\ PWn\beta VAmbs \end{Bmatrix} = {}^{Out}\Lambda^{PWn}_{PWnOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{V}^{PWn}_{PWnOutNd[\beta]} + \vec{V}^{PWn}_{PWnOutNd[\beta]+1}\right)\right\}$$

$$\begin{Bmatrix} PWn\beta STVn \\ PWn\beta STVc \\ PWn\beta STVs \end{Bmatrix} = {}^{Out}\Lambda^{PWn}_{PWnOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{v}^{PWn}_{PWnOutNd[\beta]} + \vec{v}^{PWn}_{PWnOutNd[\beta]+1}\right)\right\}$$

$$PWn\beta VRel = U_{rel_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}}$$

$$PWn\beta DynP = \frac{1}{2}\rho U^2_{rel_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}}$$

$$\begin{Bmatrix} PWn\beta\ Re \\ PWn\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}\ U_{rel_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}}}{1000000\,KinVisc} \\ \dfrac{U_{rel_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} PWn\beta VIndn \\ PWn\beta VIndc \\ PWn\beta VInds \end{Bmatrix} = {}^{Out}\Lambda^{PWn}_{PWnOutNd[\beta]} \vec{U}_{ind_{PWnOutNd[\beta]+NumFusNds-1+NumSWnNds-1}}$$

$$PWn\beta Alpha=\frac{180}{\pi}\alpha_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}$$

$$PFlp\alpha Ctrl=Ctrl^{PFlp}[\alpha]$$

$$\begin{Bmatrix} PWn\beta Cl \\ PWn\beta Cd \\ PWn\beta Cm \\ PWn\beta Cn \\ PWn\beta Cc \end{Bmatrix}=\begin{Bmatrix} C_{l_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} \\ C_{d_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} \\ C_{m_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} \\ C_{l_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} COS\left(\alpha_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}\right)+C_{d_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} SIN\left(\alpha_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}\right) \\ -C_{l_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} SIN\left(\alpha_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}\right)+C_{d_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}} COS\left(\alpha_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} PWn\beta Fl \\ PWn\beta Fd \\ PWn\beta Mm \\ PWn\beta Fn \\ PWn\beta Fc \end{Bmatrix}=\begin{Bmatrix} (PWn\beta Dyn)(PWn\beta Cl) \\ (PWn\beta Dyn)(PWn\beta Cd) \\ (PWn\beta Dyn)c_{\substack{PWnOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1}}(PWn\beta Cm) \\ (PWn\beta Dyn)(PWn\beta Cn) \\ (PWn\beta Dyn)(PWn\beta Cc) \end{Bmatrix}$$

Vertical Stabilizer

$$\begin{Bmatrix} VS\beta VAmbn \\ VS\beta VAmbc \\ VS\beta VAmbs \end{Bmatrix}={}^{Out}\Lambda^{VS}_{VSOutNd[\beta]}\left\{\frac{1}{2}\left(\vec{V}^{VS}_{VSOutNd[\beta]}+\vec{V}^{VS}_{VSOutNd[\beta]+1}\right)\right\}$$

$$\begin{Bmatrix} VS\beta STVn \\ VS\beta STVc \\ VS\beta STVs \end{Bmatrix}={}^{Out}\Lambda^{VS}_{VSOutNd[\beta]}\left\{\frac{1}{2}\left(\vec{v}^{VS}_{VSOutNd[\beta]}+\vec{v}^{VS}_{VSOutNd[\beta]+1}\right)\right\}$$

$$VS\beta VRel=U_{rel_{\substack{VSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}}}$$

$$VS\beta DynP=\frac{1}{2}\rho U^2_{rel_{\substack{VSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1}}}$$

$$\begin{Bmatrix} VS\beta\, Re \\ VS\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}} U_{rel_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}}}{1000000 KinVisc} \\ \dfrac{U_{rel_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} VS\beta VIndn \\ VS\beta VIndc \\ VS\beta VInds \end{Bmatrix} = {}^{Out}\Lambda^{VS}_{VSOutNd[\beta]} \vec{U}_{ind_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}}$$

$$VS\beta Alpha = \frac{180}{\pi} \alpha_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}$$

$$Rudr\alpha Ctrl = Ctrl^{Rudr}[\alpha]$$

$$\begin{Bmatrix} VS\beta Cl \\ VS\beta Cd \\ VS\beta Cm \\ VS\beta Cn \\ VS\beta Cc \end{Bmatrix} = \begin{Bmatrix} C_{l_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} \\ C_{d_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} \\ C_{m_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} \\ C_{l_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} COS\left(\alpha_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}\right) + C_{d_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} SIN\left(\alpha_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}\right) \\ -C_{l_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} SIN\left(\alpha_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}\right) + C_{d_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}} COS\left(\alpha_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} VS\beta Fl \\ VS\beta Fd \\ VS\beta Mm \\ VS\beta Fn \\ VS\beta Fc \end{Bmatrix} = \begin{Bmatrix} (VS\beta Dyn)(VS\beta Cl) \\ (VS\beta Dyn)(VS\beta Cd) \\ (VS\beta Dyn) c_{\substack{VSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1}} (VS\beta Cm) \\ (VS\beta Dyn)(VS\beta Cn) \\ (VS\beta Dyn)(VS\beta Cc) \end{Bmatrix}$$

Starboard (Right) Horizontal Stabilizer

$$\begin{Bmatrix} SHS\beta VAmbn \\ SHS\beta VAmbc \\ SHS\beta VAmbs \end{Bmatrix} = {}^{Out}\Lambda^{SHS}_{SHSOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{V}^{SHS}_{SHSOutNd[\beta]} + \vec{V}^{SHS}_{SHSOutNd[\beta]+1} \right) \right\}$$

$$\begin{Bmatrix} SHS\beta STVn \\ SHS\beta STVc \\ SHS\beta STVs \end{Bmatrix} = {}^{Out}\Lambda^{SHS}_{SHSOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{v}^{SHS}_{SHSOutNd[\beta]} + \vec{v}^{SHS}_{SHSOutNd[\beta]+1} \right) \right\}$$

$$SHS\beta VRel = U_{rel_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}}$$

$$SHS\beta DynP = \frac{1}{2}\rho U^2_{rel_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}}$$

$$\begin{Bmatrix} SHS\beta\, Re \\ SHS\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}} \; U_{rel_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}}}{1000000 KinVisc} \\ \dfrac{U_{rel_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} SHS\beta VIndn \\ SHS\beta VIndc \\ SHS\beta VInds \end{Bmatrix} = {}^{Out}\Lambda^{SHS}_{SHSOutNd[\beta]} \vec{U}_{ind_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}}$$

$$SHS\beta Alpha = \frac{180}{\pi}\alpha_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}$$

$$SElv\alpha Ctrl = Ctrl^{SElv}[\alpha]$$

$$\begin{Bmatrix} SHS\beta Cl \\ SHS\beta Cd \\ SHS\beta Cm \\ SHS\beta Cn \\ SHS\beta Cc \end{Bmatrix} = \begin{Bmatrix} C_{l_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} \\ C_{d_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} \\ C_{m_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} \\ C_{l_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} COS\left(\alpha_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}\right) + C_{d_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} SIN\left(\alpha_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}\right) \\ -C_{l_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} SIN\left(\alpha_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}\right) + C_{d_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}} COS\left(\alpha_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} SHS\beta Fl \\ SHS\beta Fd \\ SHS\beta Mm \\ SHS\beta Fn \\ SHS\beta Fc \end{Bmatrix} = \begin{Bmatrix} (SHS\beta Dyn)(SHS\beta Cl) \\ (SHS\beta Dyn)(SHS\beta Cd) \\ (SHS\beta Dyn)\, c_{\substack{SHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1}} (SHS\beta Cm) \\ (SHS\beta Dyn)(SHS\beta Cn) \\ (SHS\beta Dyn)(SHS\beta Cc) \end{Bmatrix}$$

Port (Left) Horizontal Stabilizer

$$\begin{Bmatrix} PHS\beta VAmbn \\ PHS\beta VAmbc \\ PHS\beta VAmbs \end{Bmatrix} = {}^{Out}\Lambda^{PHS}_{PHSOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{V}^{PHS}_{PHSOutNd[\beta]} + \vec{V}^{PHS}_{PHSOutNd[\beta]+1} \right) \right\}$$

$$\begin{Bmatrix} PHS\beta STVn \\ PHS\beta STVc \\ PHS\beta STVs \end{Bmatrix} = {}^{Out}\Lambda^{PHS}_{PHSOutNd[\beta]} \left\{ \frac{1}{2}\left(\vec{v}^{PHS}_{PHSOutNd[\beta]} + \vec{v}^{PHS}_{PHSOutNd[\beta]+1} \right) \right\}$$

$$PHS\beta VRel = U_{rel_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}}$$

$$PHS\beta DynP=\frac{1}{2}\rho U^2_{rel_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}}$$

$$\begin{Bmatrix} PHS\beta\,Re \\ PHS\beta M \end{Bmatrix}=\begin{Bmatrix} \dfrac{c_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}} U_{rel_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}}}{1000000KinVisc} \\ \dfrac{U_{rel_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} PHS\beta VIndn \\ PHS\beta VIndc \\ PHS\beta VInds \end{Bmatrix}={}^{Out}\Lambda^{PHS}_{PHSOutNd[\beta]}\vec{U}_{ind_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}}$$

$$PHS\beta Alpha=\frac{180}{\pi}\alpha_{\substack{PHSOutNd[\beta]\\+NumFusNds-1\\+NumSWnNds-1\\+NumPWnNds-1\\+NumVSNds-1\\+NumSHSNds-1}}$$

$$PElv\alpha Ctrl=Ctrl^{PElv}[\alpha]$$

$$\begin{Bmatrix} PHS\beta Cl \\ PHS\beta Cd \\ PHS\beta Cm \\ PHS\beta Cn \\ PHS\beta Cc \end{Bmatrix} = \begin{Bmatrix} C_{l_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} \\ C_{d_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} \\ C_{m_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} \\ C_{l_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} COS\left(\alpha_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}\right) + C_{d_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} SIN\left(\alpha_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}\right) \\ -C_{l_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} SIN\left(\alpha_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}\right) + C_{d_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}} COS\left(\alpha_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}}\right) \end{Bmatrix}$$

$$\begin{Bmatrix} PHS\beta Fl \\ PHS\beta Fd \\ PHS\beta Mm \\ PHS\beta Fn \\ PHS\beta Fc \end{Bmatrix} = \begin{Bmatrix} (PHS\beta Dyn)(PHS\beta Cl) \\ (PHS\beta Dyn)(PHS\beta Cd) \\ (PHS\beta Dyn)\, c_{\substack{PHSOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1}} (PHS\beta Cm) \\ (PHS\beta Dyn)(PHS\beta Cn) \\ (PHS\beta Dyn)(PHS\beta Cc) \end{Bmatrix}$$

Pylons

$$\begin{Bmatrix} SP\alpha\beta VAmbn \\ SP\alpha\beta VAmbc \\ SP\alpha\beta VAmbs \\ PP\alpha\beta VAmbn \\ PP\alpha\beta VAmbc \\ PP\alpha\beta VAmbc \end{Bmatrix} = \begin{Bmatrix} {}^{Out}\Lambda^{SPy}_{PylOutNd[\beta]}[\alpha]\left\{\frac{1}{2}\left(\vec{V}^{SPy}_{PylOutNd[\beta]}[\alpha] + \vec{V}^{SPy}_{PylOutNd[\beta]+1}[\alpha]\right)\right\} \\ {}^{Out}\Lambda^{PPy}_{PylOutNd[\beta]}[\alpha]\left\{\frac{1}{2}\left(\vec{V}^{PPy}_{PylOutNd[\beta]}[\alpha] + \vec{V}^{PPy}_{PylOutNd[\beta]+1}[\alpha]\right)\right\} \end{Bmatrix}$$

$$\left\{\begin{matrix} SP\alpha\beta STVn \\ SP\alpha\beta STVc \\ SP\alpha\beta STVs \\ PP\alpha\beta STVn \\ PP\alpha\beta STVc \\ PP\alpha\beta STVs \end{matrix}\right\} = \left\{\begin{matrix} {}^{Out}\Lambda^{SPy}_{PylOutNd[\beta]}\left[\alpha\right]\left\{\frac{1}{2}\left(\vec{v}^{SPy}_{PylOutNd[\beta]}\left[\alpha\right] + \vec{v}^{SPy}_{PylOutNd[\beta]+1}\left[\alpha\right]\right)\right\} \\ {}^{Out}\Lambda^{PPy}_{PylOutNd[\beta]}\left[\alpha\right]\left\{\frac{1}{2}\left(\vec{v}^{PPy}_{PylOutNd[\beta]}\left[\alpha\right] + \vec{v}^{PPy}_{PylOutNd[\beta]+1}\left[\alpha\right]\right)\right\} \end{matrix}\right\}$$

$$\left\{\begin{matrix} SP\alpha\beta VRel \\ PP\alpha\beta VRel \end{matrix}\right\} = \left\{\begin{matrix} U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\ U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \end{matrix}\right\}$$

$$\left\{\begin{matrix} SP\alpha\beta DynP \\ PP\alpha\beta DynP \end{matrix}\right\} = \left\{\begin{matrix} \frac{1}{2}\rho U^2_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\ \frac{1}{2}\rho U^2_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \end{matrix}\right\}$$

$$\begin{Bmatrix} SP\alpha\beta\,Re \\ SP\alpha\beta M \\ PP\alpha\beta\,Re \\ PP\alpha\beta M \end{Bmatrix} = \begin{Bmatrix} \dfrac{c_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}} \; U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}}}{1000000\,KinVisc} \\ \dfrac{U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}}}{SpdSound} \\ \dfrac{c_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}} \; U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}}}{1000000\,KinVisc} \\ \dfrac{U_{rel_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}}}{SpdSound} \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha\beta VIndn \\ SP\alpha\beta VIndc \\ SP\alpha\beta VInds \\ PP\alpha\beta VIndn \\ PP\alpha\beta VIndc \\ PP\alpha\beta VInds \end{Bmatrix} = \begin{Bmatrix} {}^{Out}\Lambda^{SPy}_{PylOutNd[\beta]}[\alpha]\vec{U}_{ind_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\ {}^{Out}\Lambda^{PPy}_{PylOutNd[\beta]}[\alpha]\vec{U}_{ind_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha\beta\,Alpha \\ PP\alpha\beta\,Alpha \end{Bmatrix} = \begin{Bmatrix} \dfrac{180}{\pi}\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}} \\ \dfrac{180}{\pi}\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}} \end{Bmatrix}$$

$$
\begin{Bmatrix} SP\alpha\beta Cl \\ SP\alpha\beta Cd \\ SP\alpha\beta Cm \\ SP\alpha\beta Cn \\ SP\alpha\beta Cc \\ PP\alpha\beta Cl \\ PP\alpha\beta Cd \\ PP\alpha\beta Cm \\ PP\alpha\beta Cn \\ PP\alpha\beta Cc \end{Bmatrix}
=
\begin{Bmatrix}
C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{m_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} COS\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}\right) + C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} SIN\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}\right) \\
-C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} SIN\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}\right) + C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}} COS\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}\right) \\
C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{m_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} \\
C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} COS\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}\right) + C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} SIN\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}\right) \\
-C_{l_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} SIN\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}\right) + C_{d_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}} COS\left(\alpha_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}\right)
\end{Bmatrix}
$$

$$\begin{Bmatrix} SP\alpha\beta Fl \\ SP\alpha\beta Fd \\ SP\alpha\beta Mm \\ SP\alpha\beta Fn \\ SP\alpha\beta Fc \\ PP\alpha\beta Fl \\ PP\alpha\beta Fd \\ PP\alpha\beta Mm \\ PP\alpha\beta Fn \\ PP\alpha\beta Fc \end{Bmatrix} = \begin{Bmatrix} (SP\alpha\beta Dyn)(SP\alpha\beta Cl) \\ (SP\alpha\beta Dyn)(SP\alpha\beta Cd) \\ (SP\alpha\beta Dyn)\,c_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)(\alpha-1)}}\,(SP\alpha\beta Cm) \\ (SP\alpha\beta Dyn)(SP\alpha\beta Cn) \\ (SP\alpha\beta Dyn)(SP\alpha\beta Cc) \\ (PP\alpha\beta Dyn)(PP\alpha\beta Cl) \\ (PP\alpha\beta Dyn)(PP\alpha\beta Cd) \\ (PP\alpha\beta Dyn)\,c_{\substack{PylOutNd[\beta] \\ +NumFusNds-1 \\ +NumSWnNds-1 \\ +NumPWnNds-1 \\ +NumVSNds-1 \\ +NumSHSNds-1 \\ +NumPHSNds-1 \\ +(NumPylNds-1)N_{Pylons} \\ +(NumPylNds-1)(\alpha-1)}}\,(PP\alpha\beta Cm) \\ (PP\alpha\beta Dyn)(PP\alpha\beta Cn) \\ (PP\alpha\beta Dyn)(PP\alpha\beta Cc) \end{Bmatrix}$$

Rotors

$$\begin{Bmatrix} SP\alpha TTSR \\ SP\alpha BTSR \\ PP\alpha TTSR \\ PP\alpha BTSR \end{Bmatrix} = \begin{Bmatrix} \left| \dfrac{\Omega^{SPyRtr}[\alpha,1]\,R^{SPyRtr}[\alpha,1]}{{}^{Rel}\vec{V}^{SPyRtr}[\alpha,1]\bullet\hat{x}^{SPyRtr}[\alpha,1]} \right| \\ \left| \dfrac{\Omega^{SPyRtr}[\alpha,2]\,R^{SPyRtr}[\alpha,2]}{{}^{Rel}\vec{V}^{SPyRtr}[\alpha,2]\bullet\hat{x}^{SPyRtr}[\alpha,2]} \right| \\ \left| \dfrac{\Omega^{PPyRtr}[\alpha,1]\,R^{PPyRtr}[\alpha,1]}{{}^{Rel}\vec{V}^{PPyRtr}[\alpha,1]\bullet\hat{x}^{PPyRtr}[\alpha,1]} \right| \\ \left| \dfrac{\Omega^{PPyRtr}[\alpha,2]\,R^{PPyRtr}[\alpha,2]}{{}^{Rel}\vec{V}^{PPyRtr}[\alpha,2]\bullet\hat{x}^{PPyRtr}[\alpha,2]} \right| \end{Bmatrix}$$

Commented [JJ31]: R^SPyRtr and R^PPyRtr are the RtrRad inputs.

If the denominator is zero, then TSR should be set to zero.

$$\begin{Bmatrix} SP\alpha TPitch \\ SP\alpha BPitch \\ PP\alpha TPitch \\ PP\alpha BPitch \end{Bmatrix} = \frac{180}{\pi} \begin{Bmatrix} \theta^{SPyRtr}[\alpha,1] \\ \theta^{SPyRtr}[\alpha,2] \\ \theta^{PPyRtr}[\alpha,1] \\ \theta^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TSkew \\ SP\alpha BSkew \\ PP\alpha TSkew \\ PP\alpha BSkew \end{Bmatrix} = \frac{180}{\pi} \begin{Bmatrix} \chi^{SPyRtr}[\alpha,1] \\ \chi^{SPyRtr}[\alpha,2] \\ \chi^{PPyRtr}[\alpha,1] \\ \chi^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TRtSpd \\ SP\alpha BRtSpd \\ PP\alpha TRtSpd \\ PP\alpha BRtSpd \end{Bmatrix} = \begin{Bmatrix} \Omega^{SPyRtr}[\alpha,1] \\ \Omega^{SPyRtr}[\alpha,2] \\ \Omega^{PPyRtr}[\alpha,1] \\ \Omega^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TVRel \\ SP\alpha BVRel \\ PP\alpha TVRel \\ PP\alpha BVRel \end{Bmatrix} = \begin{Bmatrix} \left\| {}^{Rel}\vec{V}^{SPyRtr}[\alpha,1] \right\|_2 \\ \left\| {}^{Rel}\vec{V}^{SPyRtr}[\alpha,2] \right\|_2 \\ \left\| {}^{Rel}\vec{V}^{PPyRtr}[\alpha,1] \right\|_2 \\ \left\| {}^{Rel}\vec{V}^{PPyRtr}[\alpha,2] \right\|_2 \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TCp \\ SP\alpha BCp \\ PP\alpha TCp \\ SP\alpha BCp \\ SP\alpha TCq \\ SP\alpha BCq \\ PP\alpha TCq \\ PP\alpha BCq \\ SP\alpha TCt \\ SP\alpha BCt \\ PP\alpha TCt \\ PP\alpha BCt \end{Bmatrix} = \begin{Bmatrix} C_P^{SPyRtr}[\alpha,1] \\ C_P^{SPyRtr}[\alpha,2] \\ C_P^{PPyRtr}[\alpha,1] \\ C_P^{PPyRtr}[\alpha,2] \\ -C_{Mx}^{SPyRtr}[\alpha,1] \\ -C_{Mx}^{SPyRtr}[\alpha,2] \\ -C_{Mx}^{PPyRtr}[\alpha,1] \\ -C_{Mx}^{PPyRtr}[\alpha,2] \\ -C_{Fx}^{SPyRtr}[\alpha,1] \\ -C_{Fx}^{SPyRtr}[\alpha,2] \\ -C_{Fx}^{PPyRtr}[\alpha,1] \\ -C_{Fx}^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

Deleted: $$\begin{Bmatrix} SP\alpha TCp \\ SP\alpha BCp \\ PP\alpha TCp \\ SP\alpha BCp \\ SP\alpha TCq \\ SP\alpha BCq \\ PP\alpha TCq \\ PP\alpha BCq \\ SP\alpha TCt \\ SP\alpha BCt \\ PP\alpha TCt \\ PP\alpha BCt \end{Bmatrix} = \begin{Bmatrix} C_P^{SPyRtr}[\alpha,1] \\ C_P^{SPyRtr}[\alpha,2] \\ C_P^{PPyRtr}[\alpha,1] \\ C_P^{PPyRtr}[\alpha,2] \\ C_{Mx}^{SPyRtr}[\alpha,1] \\ C_{Mx}^{SPyRtr}[\alpha,2] \\ C_{Mx}^{PPyRtr}[\alpha,1] \\ C_{Mx}^{PPyRtr}[\alpha,2] \\ -C_{Fx}^{SPyRtr}[\alpha,1] \\ -C_{Fx}^{SPyRtr}[\alpha,2] \\ -C_{Fx}^{PPyRtr}[\alpha,1] \\ -C_{Fx}^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TFx \\ SP\alpha BFx \\ PP\alpha TFx \\ PP\alpha BFx \\ SP\alpha TFy \\ SP\alpha BFy \\ PP\alpha TFy \\ PP\alpha BFy \\ SP\alpha TFz \\ SP\alpha BFz \\ PP\alpha TFz \\ PP\alpha BFz \\ SP\alpha TMx \\ SP\alpha BMx \\ PP\alpha TMx \\ PP\alpha BMx \\ SP\alpha TMy \\ SP\alpha BMy \\ PP\alpha TMy \\ PP\alpha BMy \\ SP\alpha TMz \\ SP\alpha BMz \\ PP\alpha TMz \\ PP\alpha BMz \end{Bmatrix} = \begin{Bmatrix} \vec{F}^{SPyRtr}[\alpha,1] \bullet \hat{x}^{SPyRtr}[\alpha,1] \\ \vec{F}^{SPyRtr}[\alpha,2] \bullet \hat{x}^{SPyRtr}[\alpha,2] \\ \vec{F}^{PPyRtr}[\alpha,1] \bullet \hat{x}^{PPyRtr}[\alpha,1] \\ \vec{F}^{PPyRtr}[\alpha,2] \bullet \hat{x}^{PPyRtr}[\alpha,2] \\ \vec{F}^{SPyRtr}[\alpha,1] \bullet \hat{y}^{SPyRtr}[\alpha,1] \\ \vec{F}^{SPyRtr}[\alpha,2] \bullet \hat{y}^{SPyRtr}[\alpha,2] \\ \vec{F}^{PPyRtr}[\alpha,1] \bullet \hat{y}^{PPyRtr}[\alpha,1] \\ \vec{F}^{PPyRtr}[\alpha,2] \bullet \hat{y}^{PPyRtr}[\alpha,2] \\ \vec{F}^{SPyRtr}[\alpha,1] \bullet \hat{z}^{SPyRtr}[\alpha,1] \\ \vec{F}^{SPyRtr}[\alpha,2] \bullet \hat{z}^{SPyRtr}[\alpha,2] \\ \vec{F}^{PPyRtr}[\alpha,1] \bullet \hat{z}^{PPyRtr}[\alpha,1] \\ \vec{F}^{PPyRtr}[\alpha,2] \bullet \hat{z}^{PPyRtr}[\alpha,2] \\ \vec{M}^{SPyRtr}[\alpha,1] \bullet \hat{x}^{SPyRtr}[\alpha,1] \\ \vec{M}^{SPyRtr}[\alpha,2] \bullet \hat{x}^{SPyRtr}[\alpha,2] \\ \vec{M}^{PPyRtr}[\alpha,1] \bullet \hat{x}^{PPyRtr}[\alpha,1] \\ \vec{M}^{PPyRtr}[\alpha,2] \bullet \hat{x}^{PPyRtr}[\alpha,2] \\ \vec{M}^{SPyRtr}[\alpha,1] \bullet \hat{y}^{SPyRtr}[\alpha,1] \\ \vec{M}^{SPyRtr}[\alpha,2] \bullet \hat{y}^{SPyRtr}[\alpha,2] \\ \vec{M}^{PPyRtr}[\alpha,1] \bullet \hat{y}^{PPyRtr}[\alpha,1] \\ \vec{M}^{PPyRtr}[\alpha,2] \bullet \hat{y}^{PPyRtr}[\alpha,2] \\ \vec{M}^{SPyRtr}[\alpha,1] \bullet \hat{z}^{SPyRtr}[\alpha,1] \\ \vec{M}^{SPyRtr}[\alpha,2] \bullet \hat{z}^{SPyRtr}[\alpha,2] \\ \vec{M}^{PPyRtr}[\alpha,1] \bullet \hat{z}^{PPyRtr}[\alpha,1] \\ \vec{M}^{PPyRtr}[\alpha,2] \bullet \hat{z}^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

$$\begin{Bmatrix} SP\alpha TPwr \\ SP\alpha BPwr \\ PP\alpha TPwr \\ PP\alpha BPwr \end{Bmatrix} = \begin{Bmatrix} P^{SPyRtr}[\alpha,1] \\ P^{SPyRtr}[\alpha,2] \\ P^{PPyRtr}[\alpha,1] \\ P^{PPyRtr}[\alpha,2] \end{Bmatrix}$$

Energy Kite

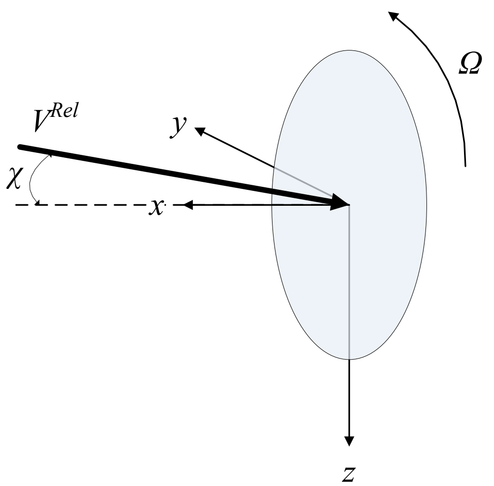
$$\begin{Bmatrix} KiteFxi \\ KiteFyi \\ KiteFzi \\ KiteMxi \\ KiteMyi \\ KiteMzi \end{Bmatrix} = \left\{ \begin{array}{l} M_F^{P2P}\left(\vec{F}_j^{Fus}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{SWn}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{PWn}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{VS}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{SHS}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{PHS}\right) \\ +M_F^{P2P}\left(\vec{F}_j^{SPy}\left[n_{Pylons}\right]\right) \\ +M_F^{P2P}\left(\vec{F}_j^{PPy}\left[n_{Pylons}\right]\right) \\ +M_F^{P2P}\left(\vec{F}^{SPyRtr}\left[n_{Pylons},n_2\right]\right) \\ +M_F^{P2P}\left(\vec{F}^{PPyRtr}\left[n_{Pylons},n_2\right]\right) \\ M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{Fus},\vec{F}_j^{Fus},\vec{M}_j^{Fus}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{SWn},\vec{F}_j^{SWn},\vec{M}_j^{SWn}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{PWn},\vec{F}_j^{PWn},\vec{M}_j^{PWn}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{VS},\vec{F}_j^{VS},\vec{M}_j^{VS}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{SHS},\vec{F}_j^{SHS},\vec{M}_j^{SHS}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{PHS},\vec{F}_j^{PHS},\vec{M}_j^{PHS}\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{SPy}\left[n_{Pylons}\right],\vec{F}_j^{SPy}\left[n_{Pylons}\right],\vec{M}_j^{SPy}\left[n_{Pylons}\right]\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},{}^{Out}\vec{u}_j^{PPy}\left[n_{Pylons}\right],\vec{F}_j^{PPy}\left[n_{Pylons}\right],\vec{M}_j^{PPy}\left[n_{Pylons}\right]\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},\vec{u}_j^{SPyRtr}\left[n_{Pylons},n_2\right],\vec{F}^{SPyRtr}\left[n_{Pylons},n_2\right],\vec{M}^{SPyRtr}\left[n_{Pylons},n_2\right]\right) \\ +M_M^{P2P}\left(\vec{u}^{FusO},\vec{u}_j^{PPyRtr}\left[n_{Pylons},n_2\right],\vec{F}^{PPyRtr}\left[n_{Pylons},n_2\right],\vec{M}^{PPyRtr}\left[n_{Pylons},n_2\right]\right) \end{array} \right\}$$

$$KitePwr = P^{SPyRtr}\left[n_{Pylons},n_2\right]+P^{PPyRtr}\left[n_{Pylons},n_2\right]$$

Actuator Disk (MODULE ActuatorDisk)

Solves quasi-steady actuator disk to calculate the 3 forces/3 moments/power dependent on the rotor speed (Ω), rotor inflow relative wind speed (vector magnitude V^{Rel}), rotor inflow skew angle (χ), and rotor-collective blade-pitch angle (θ).

Note: The actuator disk is defined with local x normal to the disk (pointed forward, in the primary direction of flight) and positive rotation (Ω) about positive local x. The V^{Rel} vector is always in the local x-y plane, and unless χ is 0˚ or 180˚, the V^{Rel} vector has a component along negative local y. Local z follows the right-hand rule. (That is, the local coordinate system rotates with the V^{Rel} vector.)



Inputs	Outputs	States	Parameters
• Ω – Rotor speed (rad/s) • V^{Rel} – Rotor inflow relative wind speed (ambient + kite motion) (vector magnitude) (m/s) • χ – Rotor inflow-skew angle (rad) • θ – Rotor-collective blade-pitch angle (rad)	• F_x – Thrust (x/axial) force (N) • F_y – Transverse (y) force (N) • F_z – Transverse (z) force (N) • M_x – Torque (x) (Nm) • M_y – Transverse (y) moment (Nm) • M_z – Transverse (z) moment (Nm) • P – Power (W)		• D – Rotor diameter (m) • N_{RtSpd} – Number of rotor speeds in tables (-) • $N_{V\,Rel}$ – Number of rotor inflow wind speeds in tables (-) • N_{χ} – Number of rotor inflow-skew angles in tables (-) • N_{θ} – Number of rotor-collective blade-pitch angles in tables (-) • $RtSpd\left[n_{RtSpd}\right]$ – Rotor speeds in tables (rad/s) • $V\,Rel\left[n_{V\,Rel}\right]$ – Rotor inflow wind speeds in tables

Deleted: R

Deleted: radius

Deleted: <#> $\frac{1}{2}\rho A$ – ½ air density times rotor swept area (kg/m)¶

			(m/s) • $\chi[n_\chi]$ – Rotor inflow-skew angles in tables (rad) • $\theta[n_\theta]$ – Rotor-collective blade-pitch angles in tables (rad) • $C_{Fx}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Thrust (x/axial) force coefficient (-) • $C_{Fy}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Transverse (y) force coefficient (-) • $C_{Fz}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Transverse (z) force coefficient (-) • $C_{Mx}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Torque (x) coefficient (-) • $C_{My}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Transverse (y) moment coefficient (-) • $C_{Mz}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Transverse (z) moment coefficient (-) • $C_P[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$ – Power coefficient (-)

Initialization:

Initialization Inputs	Initialization Outputs
• R – Rotor radius (m) • ρ – Air density (kg/m^3) • $FileName$ – File name (including path) of the actuator disk input file (string)	

Set parameters from initialization inputs ($D = 2R$)

Read in parameters from $FileName$ (N_{RtSpd}, $N_{V\,Rel}$, N_χ, N_θ, $RtSpd[n_{RtSpd}]$, $V\,Rel[n_{V\,Rel}]$, $\chi[n_\chi]$, $\theta[n_\theta]$, $C_{Fx}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_{Fy}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_{Fz}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_{Mx}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_{My}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_{Mz}[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$, $C_P[n_{RtSpd}, n_{V\,Rel}, n_\chi, n_\theta]$).

Convert units of $\theta[n_\theta]$ and $\chi[n_\chi]$ from degrees to radians.

Deleted: R

Deleted: , $\frac{1}{2}\rho A = \frac{1}{2}\rho\pi R^2$

Note/verify restrictions on input data:

- $0 < R$.
- $0 < \rho$.
- $2 \leq N_{RtSpd}$; must be the same for each table; note: $1 \leq n_{RtSpd} \leq N_{RtSpd}$.
- $2 \leq N_{V\,Rel}$; must be the same for each table; note: $1 \leq n_{V\,Rel} \leq N_{V\,Rel}$.
- $2 \leq N_{\chi}$; must be the same for each table; note: $1 \leq n_{\chi} \leq N_{\chi}$.
- $2 \leq N_{\theta}$; must be the same for each table; note: $1 \leq n_{\theta} \leq N_{\theta}$.
- $RtSpd\left[n_{RtSpd}\right]$; data must be entered monotonically increasing.
- $0 \leq V\,Rel\left[n_{V\,Rel}\right]$; data must be entered monotonically increasing.
- $0 \leq \chi\left[n_{\chi}\right] \leq \pi$ (radians); data must be entered monotonically increasing.
- $-\pi \leq \theta\left[n_{\theta}\right] \leq \pi$ (radians); data must be entered monotonically increasing.

Commented [JJ32]: Don't need to have lower and upper bounds on theta.

Update States:
Blank.

Calculate Outputs:

Trigger a fatal error if: $\left(\left(\Omega < RtSpd[1]\right).OR.\left(\Omega > RtSpd\left[N_{RtSpd}\right]\right)\right)$, $\left(\left(V^{Rel} < V\,Rel[1]\right).OR.\left(V^{Rel} > V\,Rel\left[N_{V\,Rel}\right]\right)\right)$, $\left(\left(\chi < \chi[1]\right).OR.\left(\chi > \chi\left[N_{\chi}\right]\right)\right)$, or $\left(\left(\theta < \theta[1]\right).OR.\left(\theta > \theta\left[N_{\theta}\right]\right)\right)$.

Deleted: Calculate the local rotor inflow wind speed, normal to the disk (vector magnitude): $V_x^{Rel} = \left|V^{Rel} COS(\chi)\right|$¶

Calculate the outputs via 4D interpolation:

Commented [jmj33]: I'd prefer this to use the 4D isoparametric interpolation that is currently used in InflowWind

$$F_x = INTERP4D\left(\rho D^4\left(\frac{\Omega}{2\pi}\right)^2 C_{Fx}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

$$F_y = INTERP4D\left(\rho D^4\left(\frac{\Omega}{2\pi}\right)^2 C_{Fy}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

$$F_z = INTERP4D\left(\rho D^4\left(\frac{\Omega}{2\pi}\right)^2 C_{Fz}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

$$M_x = INTERP4D\left(\rho D^5\left(\frac{\Omega}{2\pi}\right)^2 C_{Mx}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

$$M_y = INTERP4D\left(\rho D^5\left(\frac{\Omega}{2\pi}\right)^2 C_{My}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

$$M_z = INTERP4D\left(\rho D^5\left(\frac{\Omega}{2\pi}\right)^2 C_{Mz}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$

Deleted: $F_x = INTERP4D\left(\frac{1}{2}\rho A\left(V_x^{Rel}\right)^2 C_{Fx}[:,:,$

$F_y = INTERP4D\left(\frac{1}{2}\rho A\left(V_x^{Rel}\right)^2 C_{Fy}[:,:,:,:], RtS$

$F_z = INTERP4D\left(\frac{1}{2}\rho A\left(V_x^{Rel}\right)^2 C_{Fz}[:,:,:,:], RtS$

$M_x = INTERP4D\left(\frac{1}{2}\rho AR\left(V_x^{Rel}\right)^2 C_{Mx}[:,:,:,:],$

$M_y = INTERP4D\left(\frac{1}{2}\rho AR\left(V_x^{Rel}\right)^2 C_{My}[:,:,:,:],$

$M_z = INTERP4D\left(\frac{1}{2}\rho AR\left(V_x^{Rel}\right)^2 C_{Mz}[:,:,:,:],$

$P = INTERP4D\left(\frac{1}{2}\rho A\left(V_x^{Rel}\right)^3 C_P[:,:,:,:], RtSp$

¶

$$P = INTERP4D\left(\rho D^{5}\left(\frac{\Omega}{2\pi}\right)^{3} C_{P}[:,:,:,:], RtSpd[:]@\Omega, V\,Rel[:]@V^{Rel}, \chi[:]@\chi, \theta[:]@\theta\right)$$